

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Captura y análisis de datos en Twitter
Capture and analysis of data on Twitter

Realizado por
José Gómez Baco
Tutorizado por
Carlos Rossi Jiménez
Manuel Enciso García-Oliveros
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, septiembre - 2018

Fecha defensa:
El Secretario del Tribunal

Resumen: El proyecto consiste en la creación de una aplicación web que permita realizar la captura y análisis de datos en Twitter facilitando el estudio tanto de las opiniones expresadas como de las personas o entidades que han influido en dichas opiniones mediante técnicas de Visual Analytics.

Dicho objetivo tiene especial importancia debido al gran número de usuarios de esta red social y la inmensa cantidad de información accesibles sobre multitud de ámbitos.

El procesamiento de los datos obtenidos se centrará en el estudio de la difusión y propagación de tendencias, la búsqueda de relaciones entre usuarios en base a las acciones realizadas por estos (respuestas, citas y menciones) y el análisis del comportamiento de los usuarios usando como métricas su actividad e impacto.

Para ello el análisis se apoyará en el entorno y lenguaje de programación “R”, que provee un gran conjunto de funciones estadísticas y de cálculo numérico que permite el manejo de grandes volúmenes de datos, además de herramientas estadísticas adecuadas para la generación de gráficas.

Todo ello estará integrado en una aplicación desarrollada en el lenguaje de programación Python y bajo el framework Django, en la cual se han integrado las herramientas del lenguaje R para el procesado de los datos mediante scripts que son invocados desde la propia aplicación.

Para asegurar la integridad de la aplicación, así como de los resultados, se han limitado las posibles acciones definiendo diversos perfiles de usuarios en función de las funciones a realizar por cada uno de ellos.

Palabras claves: Twitter, Análisis, Visual Analytics, R, Red social, Python, Django

Abstract: The project consists on the creation of a web application that allows the capture and analysis of data on Twitter, facilitating the study of the opinions expressed as well as the people or entities that have influenced these opinions through Visual Analytics techniques.

This objective is especially important due to the large number of users of this social network and the immense amount of information available on a multitude of areas.

The processing of the obtained data will focus on the study of the diffusion and propagation of tendencies, the search of relations between users based on actions carried out by them (answers, quotes and mentions) and the analysis of the behavior of the users using as metrics their activity and impact.

For this, the analysis will be based on the "R" programming environment and language, which provides a large set of statistical and numerical calculation functions that allow the handling of large volumes of data, in addition to statistical tools suitable for the generation of graphs.

All this will be integrated into an application developed with Python programming language and Django framework, in which the tools of the R language have been integrated for the processing of data through scripts that are invoked from the application.

To ensure the integrity of the application, as well as of results, the possible actions have been limited by different user profiles depending on the functions that each of them can carry out.

Keywords: Twitter, Analytics, Visual Analytics, R, Social Network, Python, Django

Tabla de contenido

1. Introducción	5
1.1 Introducción	6
1.2 Visual Analytics	9
1.3 Objetivos	12
2. Análisis de datos	13
2.1 Introducción	14
2.2 Tendencias	14
2.3 Evolución	16
2.4 Relaciones entre usuarios	16
2.5 Sentimientos	17
2.6 Usuarios	18
2.7 Tendencias del momento	19
3. Análisis	21
3.1 Requisitos	22
3.1.1 Requisitos funcionales	22
3.1.2 Requisitos no funcionales	23
3.1.3 Requisitos de información	24
3.2 Casos de uso	26
4. Diseño	31
4.1 Estructuras de datos	32
4.1.1 Modelo lógico	32
4.1.2 Modelo relacional	36
4.2 Estructura del proyecto	38
4.3 Metodología de desarrollo	41
4.4 Funcionamiento del sistema	42
4.4.1 Django	42
4.4.2 R	48
5. Tecnologías	51
5.1 Tecnologías Web	52
5.1.1 HTML	52
5.1.2 Javascript	52
5.1.3 CSS	53

5.2	Lenguajes de programación / Script.....	53
5.2.1	Python.....	53
5.2.2	R	54
5.3	Persistencia.....	56
5.3.1	PostgreSQL	56
5.4	Framework	57
5.4.1	Django	57
5.5	Integración.....	58
6.	Conclusiones	61
7.	Bibliografía.....	63
7.1	Referencias bibliográficas	64
7.2	Referencias web.....	64
8.	Anexo	65
	Manual de usuario	67

1. Introducción

1.1	Introducción	6
1.2	Visual Analytics	9
1.3	Objetivos	12

Sinopsis:

En este capítulo se realizará una introducción al proyecto resaltando los objetivos principales y su motivación.

1.1 Introducción

El concepto de red social a través de internet se puede definir como un conjunto de personas o entidades que comparten alguna relación u objetivo común y que se comunican a través de la red.

Este uso de las redes sociales se ha ido incrementando en los últimos años, siendo actualmente utilizadas por el 85% de las personas de entre 16 y 65 años con acceso a internet en España, alcanzando un total de 25.5 millones de personas en 2018, datos obtenidos del Estudio anual de redes sociales (IAB, 2018).

A continuación, se muestra una gráfica con la evolución del uso de las redes sociales sobre el total de personas que acceden a internet habitualmente en España desde el 2009 hasta la actualidad.

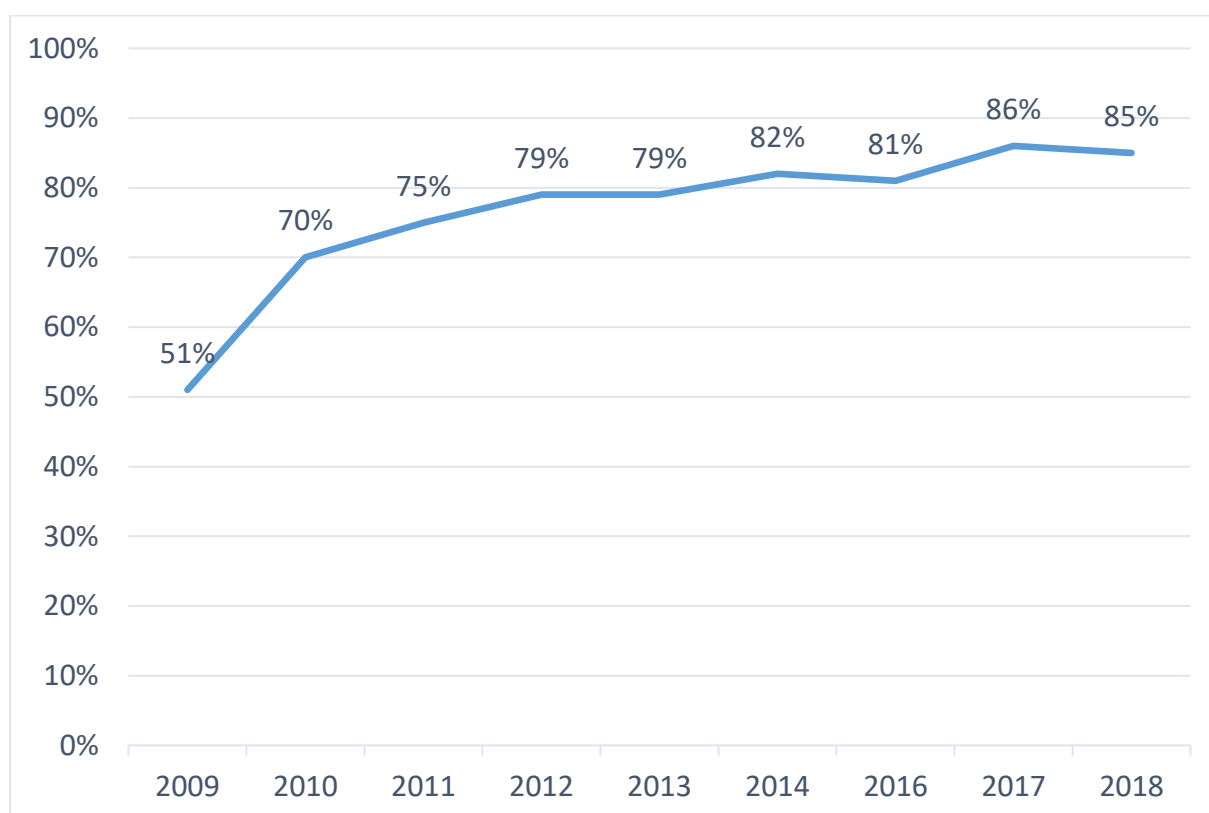


Figura 1: Evolución penetración redes sociales en España (IAB, 2018).

Este uso masivo proporciona a las redes sociales una inmensa cantidad de información sobre multitud de ámbitos diferentes debido, en gran parte, a la diversidad tanto de las redes sociales actuales como de sus propios usuarios. En consecuencia,

las redes sociales constituyen una fuente de datos realmente útil y su posterior análisis supone un problema complejo tanto en su extracción como en el análisis en sí mismo.

Un ejemplo de explotación de estos datos es el análisis de las tendencias actuales, un trabajo de gran importancia y aplicación para el comercio, el turismo, la sociología, etc.

Actualmente existen multitud de redes sociales disponibles con temáticas muy diferentes. Entre las que destacan por el porcentaje de usuarios sobre el total de los que utilizan redes sociales: Facebook, WhatsApp, Youtube, Instagram y Twitter.

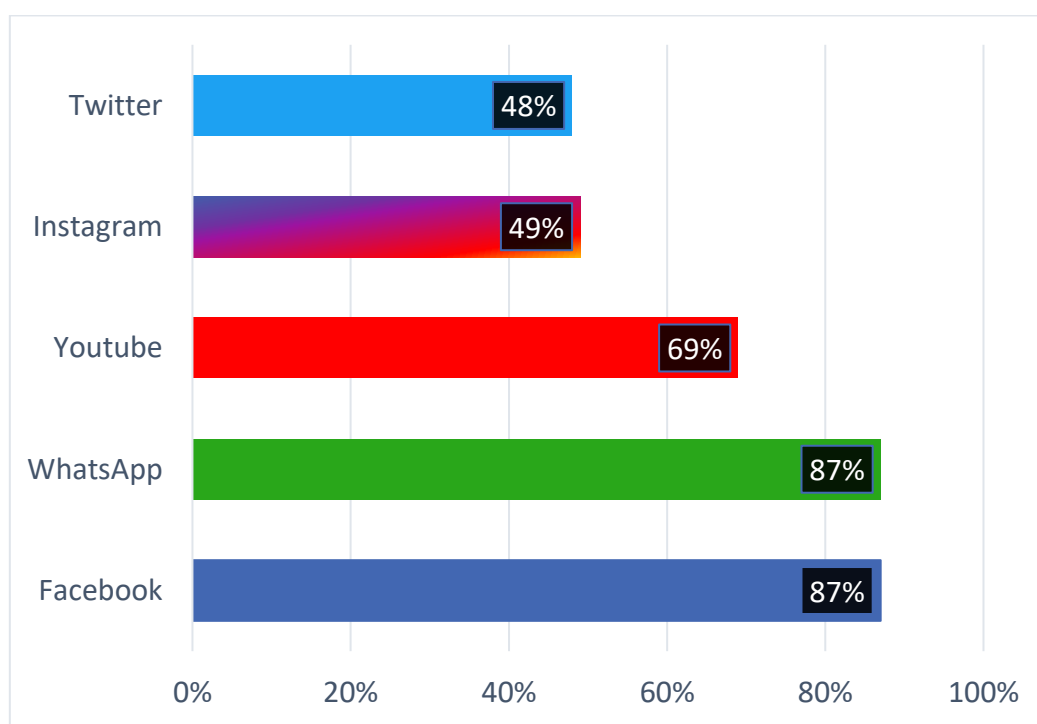


Figura 2: Porcentaje de usuarios de redes sociales (IAB, 2018).

La importancia de Twitter entre las redes sociales viene dada tanto por su número de usuarios, como por la filosofía de la red social, ya que parte de la idea del envío de mensajes cortos que son utilizados para dar a conocer noticias y expresar opiniones de diversa índole. Este uso hace que Twitter contenga información de noticias de actualidad, así como de la reacción de los usuarios ante ellas.

La característica principal por la que se ha elegido a twitter es su accesibilidad, siendo una red social que mantiene la información con carácter público y además provee de una serie de APIs (Interfaz de Programación de Aplicaciones) con la que se pueden obtener dichos datos.

El siguiente grafico muestra la evolución del número de usuarios de Twitter a nivel mundial que, a pesar no ser la red social con mayor número de usuarios, tiene una tendencia positiva.

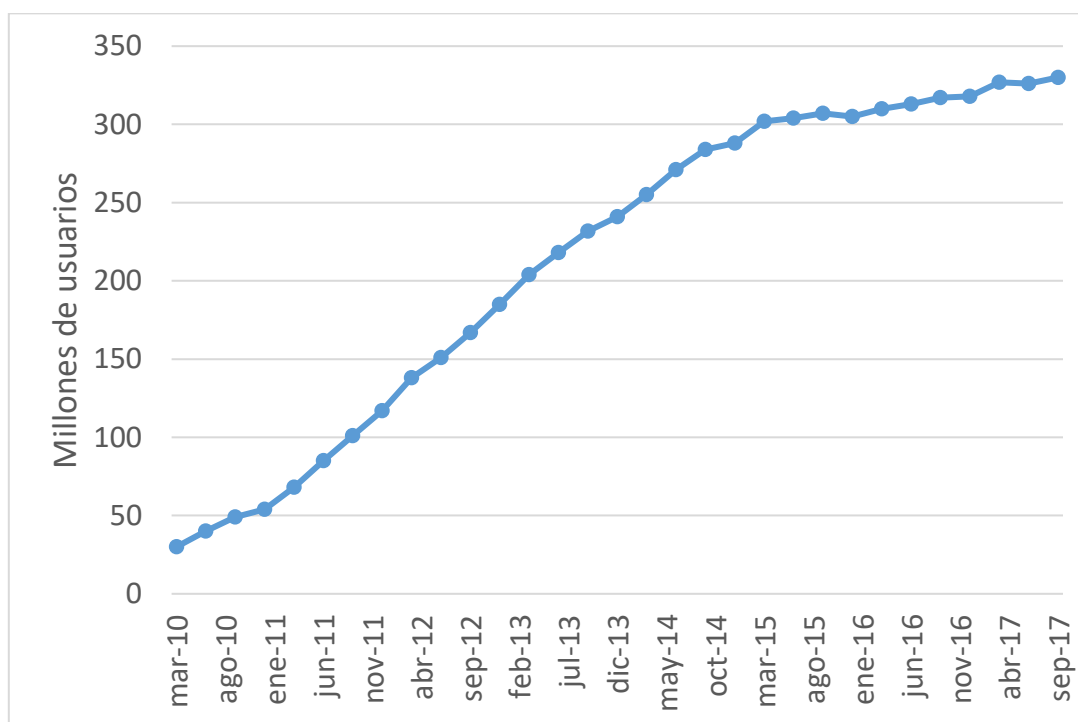


Figura 3: Evolución del uso de la red social Twitter a nivel mundial. (ABC)

Sin embargo, la información proporcionada por Twitter es desestructurada y compleja requiriendo procesamiento previo para extraer la información más relevante. Para ello, proporcionaremos una aplicación que permita este procesamiento y su posterior visualización mediante técnicas de Visual Analytics.

Para el desarrollo de esta aplicación se ha tenido en cuenta el trabajo desarrollado por D^a. Mariluz Congosto, doctora en Telemática por la Universidad Carlos III, en su tesis Caracterización de usuarios y propagación de mensajes en Twitter en el entorno de temas sociales. (Congosto, 2016)

Otro trabajo de referencia es la plataforma t-hoarder desarrollada por D^a. Mariluz Congosto que tiene como función la captura y visualización de datos obtenidos de la red social Twitter, en la que recoge la participación de los usuarios en relación a diversos temas.

1.2 Visual Analytics

Hoy en día los avances tecnológicos han hecho que la cantidad de información disponible crezca a una velocidad que hace que sea difícil de mantener y explotar. Esta información se produce en prácticamente todos los ámbitos, tanto profesional como personal.

Sin embargo, la mayoría de las veces, estos datos no tienen ningún valor en sí mismos, debido a que son muchas veces almacenados sin un filtrado previo.

Otros problemas de esta sobrecarga de datos son la pérdida de tiempo y dinero, al no ser una información aprovechada en su totalidad. Por este motivo se hace necesario el uso de técnicas que permitan explotar todo ese potencial.

Visual Analytics hace referencia a una reciente área de investigación que tiene como finalidad permitir explotar cantidades masivas de información combinando la potencia del análisis computacional y las capacidades de percepción visual y análisis del ser humano.

Esta naturaleza multidisciplinar hace que sea un área difícil de definir. Sin embargo, según la práctica actual, una definición podría ser: “Visual Analytics combina técnicas de análisis automático con visualizaciones interactivas para una comprensión efectiva, razonamiento y toma de decisiones sobre la base de conjuntos de datos muy grandes y complejos”. (Kohlhammer, 2011)

El proceso de Visual Analytics combina métodos de análisis visuales y automatizados a través de la interacción humana para extraer el conocimiento. En la figura siguiente se muestra un resumen general de las diferentes etapas (representadas con elipses) y sus transiciones (flechas) en el análisis de un proceso. (Kohlhammer, 2011)

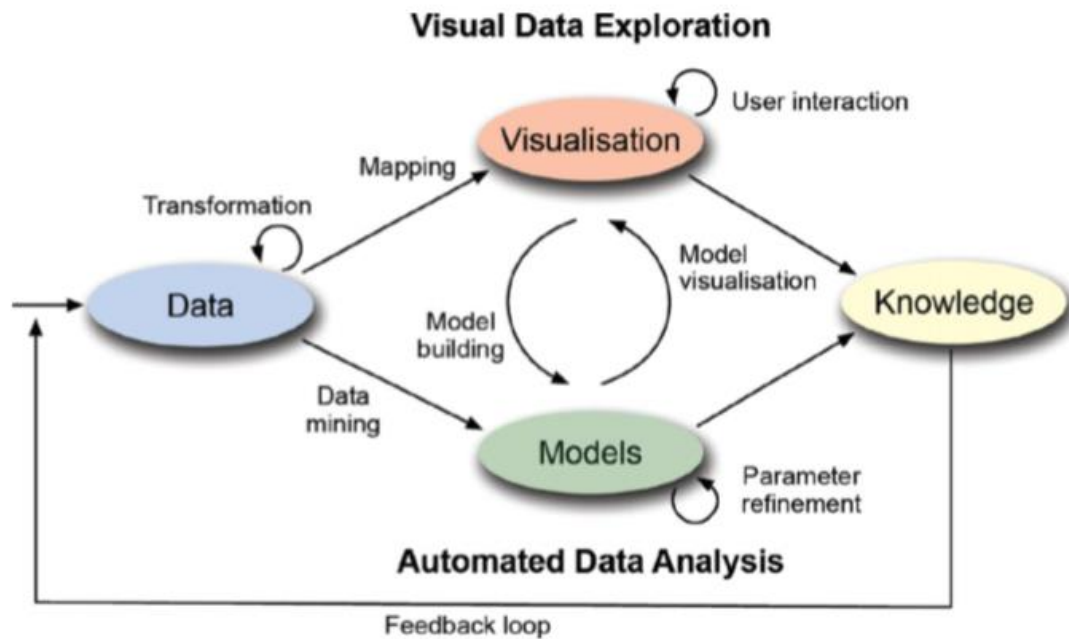


Figura 4: Procesos de Visual Analytics. (Kohlhammer, 2011)

El proceso de Visual Analytics suele llevarse a cabo en escenarios con fuentes de datos tan heterogéneas como bases de datos privadas, archivos públicos o incluso información obtenida mediante técnicas como el Scraping. En esta fase inicial, denominada descubrimiento o adquisición, dicha información deben ser integrada antes de aplicar los análisis correspondientes.

Esto implica que el siguiente paso debe ser la limpieza, que abarca tanto transformación de los datos como tareas de limpieza de datos, normalización y agrupamiento.

Después de la transformación inicial, se podrá aplicar un análisis visual o automático. En el caso del análisis automático serán aplicadas técnicas de minería de datos. Por su parte, los análisis visuales permitirán que sea el analista el que evalúe y refine el modelo, ya sea modificando parámetros o seleccionando otro algoritmo de análisis.

Finalmente se encuentra la fase de publicación, en la que se utilizan herramientas que permiten la representación gráfica e interactiva de los datos obtenidos. Dentro de esta fase encontramos herramientas de visualización estática, herramientas de visualización animada y visualizaciones web interactivas. (McGhee, 2011)

En este punto conviene aclarar que algunos autores establecen a la visualización de datos un papel en dos grandes usos: el declarativo (donde se quiere afirmar una hipótesis o ponerla de manifiesto) y el exploratorio (donde se muestran los datos para dejar que el usuario descubra el conocimiento) (Berinato 2016).

La cita es ésta: Scott Berinato Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations. Harvard Business Review Press. 2016.

En particular las principales características de las técnicas de Visual Analytics son:

- La gestión de grandes cantidades de datos que requieren tanto de computación como de participación humana.
- El objetivo final es la de extraer el mayor conocimiento posible de la información, por este motivo es necesaria la combinación de recursos humanos y tecnológicos. Por un lado, la capacidad de observación y deducción humana y por otro, herramientas como la minería de datos para la extracción automática de información.
- Esta interacción, hace que el Visual Analytics fomente el desarrollo de técnicas de minería de datos, representación y visualización de datos e informes analíticos
- Visual Analytics también integra elementos como la cognición humana, la percepción de la información, ciencias de la computación y diseño gráfico.
- Combina métodos de visualización de la información y representaciones gráficas donde la visualización incrementa la percepción humana de la siguiente manera:
 - El uso de recursos permite al usuario memorizar menos.
 - Reduciendo la búsqueda, por ejemplo, representando muchos datos en un espacio pequeño.
 - Mejorando el reconocimiento de patrones, organizando la información de manera estructurada.
 - Permitiendo la fácil inferencia de relaciones.
 - Proveyendo técnicas para la monitorización dinámica de datos.

(Kashnitsky, 2014)

1.3 **Objetivos**

El principal objetivo de este trabajo fin de grado es el desarrollo de una aplicación web que permita recuperar la información procedente de la red social Twitter, procesarla y almacenarla para su posterior análisis.

La capacidad para la visualización de datos y depurar el análisis mediante técnicas de Visual Analytics serán las características fundamentales en dicha aplicación.

El análisis de los datos obtenidos se centrará en:

- Analizar la difusión de la información
- Analizar la propagación de tendencias
- Analizar el comportamiento de los usuarios
 - Actividad
 - Impacto
- Buscar relaciones entre los usuarios (segmentación) en base a las siguientes acciones:
 - Respuestas
 - Citas
 - Menciones

2. Análisis de datos

2.1	Introducción.....	14
2.2	Tendencias.....	14
2.3	Evolución.....	16
2.4	Relaciones entre usuarios	16
2.5	Sentimientos.....	17
2.6	Usuarios	18
2.7	Tendencias del momento	19

Sinopsis: Seguidamente se detallarán los diferentes análisis que realiza el sistema, así como sus respectivas representaciones gráficas y su significado.

2.1 Introducción

En el proceso de extracción de datos (Data extraction) se ha recogido información proporcionada por la API de Twitter. En particular los relativos a los Tweets (publicaciones de los usuarios a través de la red social) con relación al término que se quiere analizar y de los usuarios que los publicaron.

Dichos datos son obtenidos con caracteres y formatos poco adecuados para su tratamiento, por lo que requieren un procesado inicial (Data cleaning) que los filtre y permita su almacenamiento para su posterior análisis. En concreto se limpian los caracteres extraños de campos de texto en los tweets y se modifica el tipo de algunos campos para su correcto almacenamiento.

Finalmente se realiza el análisis de los datos (Data Analysis), y se presentan al usuario (Data Visualization).

A continuación se presentan los diferentes análisis visuales generados.

2.2 Tendencias

La siguiente gráfica representa el número de apariciones de una palabra en los diferentes tweets, con el objetivo de exponer los términos con mayor relevancia en el estudio realizado.

Se considera que el diagrama de barras es el más apropiado para este análisis debido a que los términos analizados no mantienen relación entre ellos, representando en columnas el número de apariciones de cada termino.

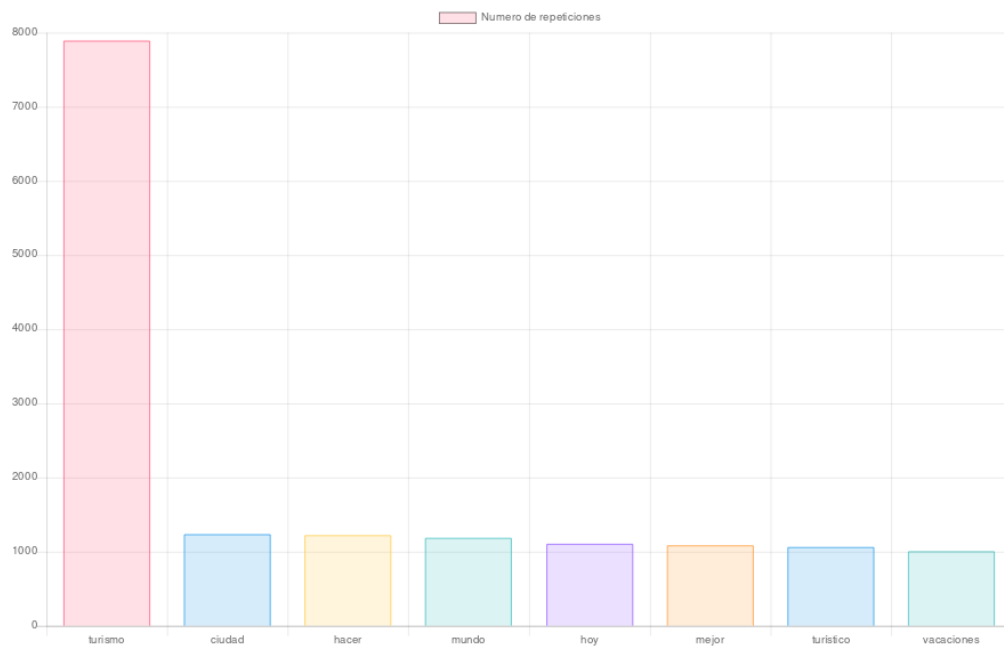


Figura 5: Gráfica de tendencias.

Se incluye una nube de palabras, en esta representación gráfica se resalta los términos más importantes mediante el tamaño y la ubicación siendo los términos más relevantes los de mayor tamaño y situados en el centro.

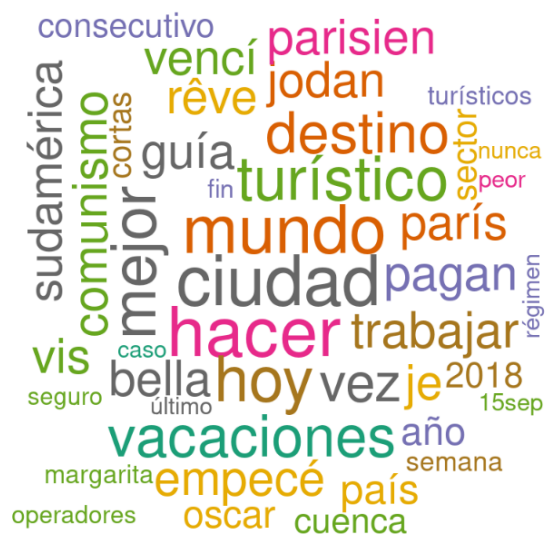


Figura 6: Nube de palabras.

2.3 Evolución

Este análisis se centra en la participación de los usuarios sobre un tema concreto a lo largo del tiempo. Siendo adecuados para su representación tanto un diagrama de líneas como uno de áreas. Se seleccionó este último porque transmite más la idea del volumen de datos alcanzado.

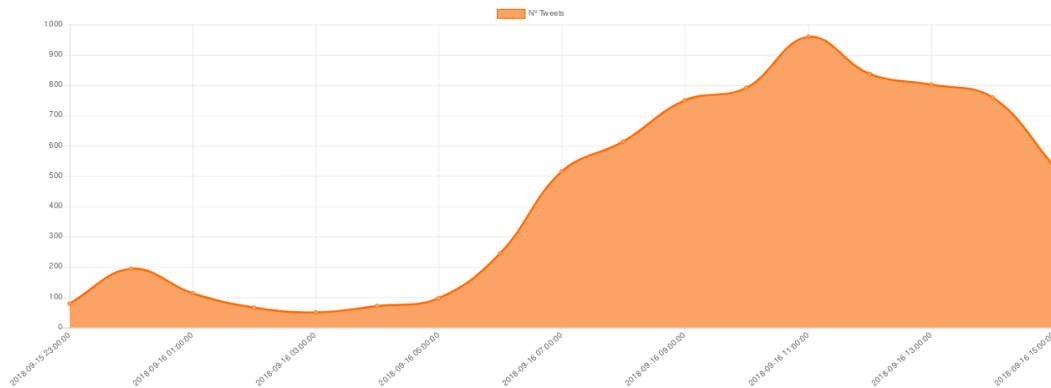


Figura 7: Gráfica de evolución.

2.4 Relaciones entre usuarios

Representa, mediante un grafo de red interactivo, las comunidades de usuarios según grado de conexión o interrelación entre los mismos. Representando mediante nodos a los usuarios, y sus relaciones mediante aristas. Estas relaciones pueden ser de menciones, respuestas y retweets.

En la representación se resaltan los nodos con mayor grado (número de aristas incidentes) mediante un tamaño superior y color rojo.

La interactividad del componente de visualización permite seleccionar los diferentes nodos, con el fin de resaltar sus relaciones directas.

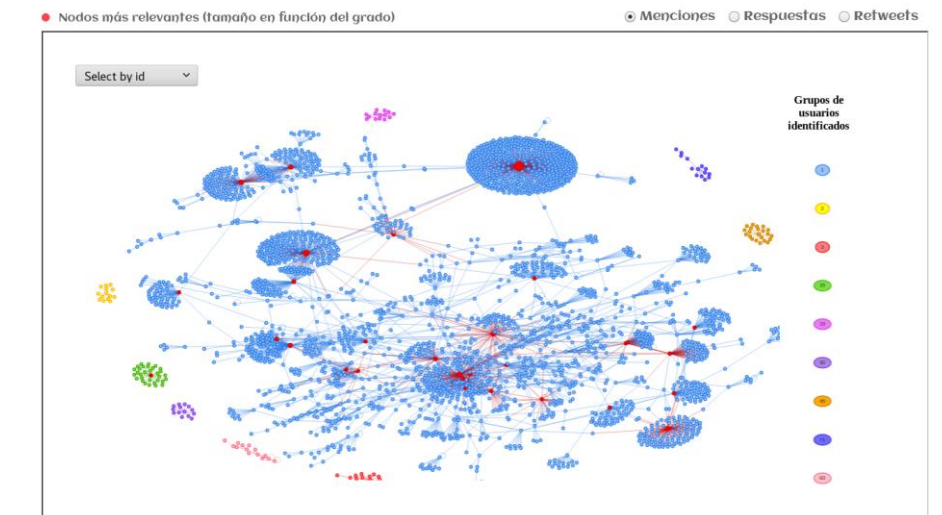


Figura 8: Gráfica de evolución.

2.5 Sentimientos

Representa, mediante un grafo interactivo, los sentimientos expresados por los diferentes usuarios mediante colores. Dichos sentimientos son calculados sobre el total de los tweets expresados por cada usuario, distinguiendo aquellos que son positivos, negativos o neutros mediante el color, como se expone en la leyenda.

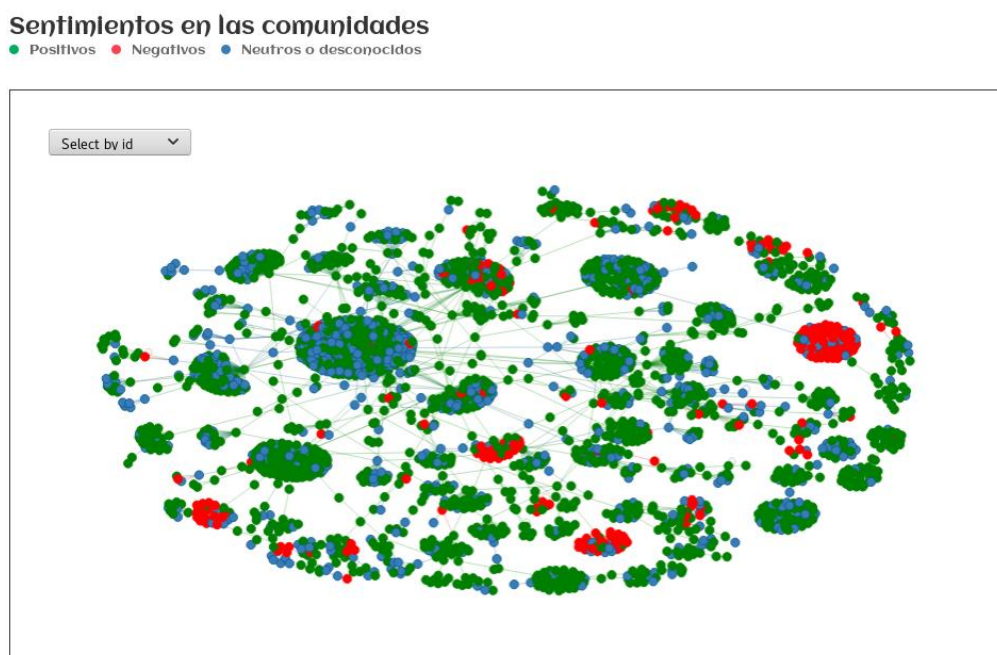


Figura 9: Gráfica de sentimientos.

Se representa el porcentaje de distribución de los sentimientos analizados mediante la siguiente gráfica de tarta, que a pesar de ser un tipo de diagrama criticado actualmente por sus limitaciones a la hora de comunicar visualmente la información, en este caso se justifica por la necesidad de comparar los caracteres positivos y negativos del sentimiento expresado.

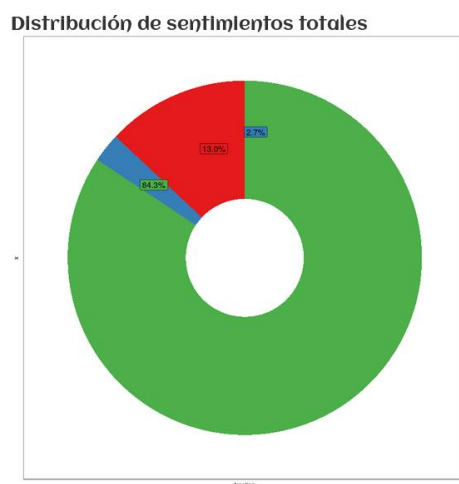


Figura 10: Gráfica de distribución de sentimientos.

2.6 Usuarios

Muestra los datos obtenidos de los distintos usuarios de forma exhaustiva, con el fin de que pueda ser explorada por el usuario. Se añade una columna con el resultado de la clasificación del usuario en base al tipo de participación en la comunicación.

Busqueda:

Nombre	Seguidores	Tweets	Seguendo	Me gusta	Creación de la cuenta	Clasificación
02_M0TIEL	196	17,512	398	781	2011-09-09 14:39:05	Común
02M0XIHERRE0	80	8,229	217	7,001	2015-07-09 20:49:01	Común
04VL0DIMIR89	23	388	39	6,494	2014-02-06 16:11:09	Común
05GUILLEM	22	512	49	1,588	2012-12-13 12:06:35	Común
05MIC0SU0REZ19	382	8,598	298	9,927	2013-06-26 23:32:54	Común
082600TIST0	0	4	2	0	2018-09-17 00:24:36	0islado
08FEPO	30	2,920	147	1,460	2013-07-17 07:06:01	Común
09340CC41600467	39	300	368	510	2014-11-10 21:58:09	Común
0__FR0NCISCO_0	146	1,947	402	1,190	2015-04-14 06:21:43	Replicador
0BOOTSESP	7	16	363	11	2018-09-17 21:25:02	Refutador

Mostrando desde 1 a 10 de 10,207 entradas

Anterior 1 2 3 4 5 ... 1021 Siguiente

Figura 11: Tabla de usuarios.

2.7 Tendencias del momento

Recoge las tendencias a nivel mundial y nacional en el momento en el que se realizó el análisis.

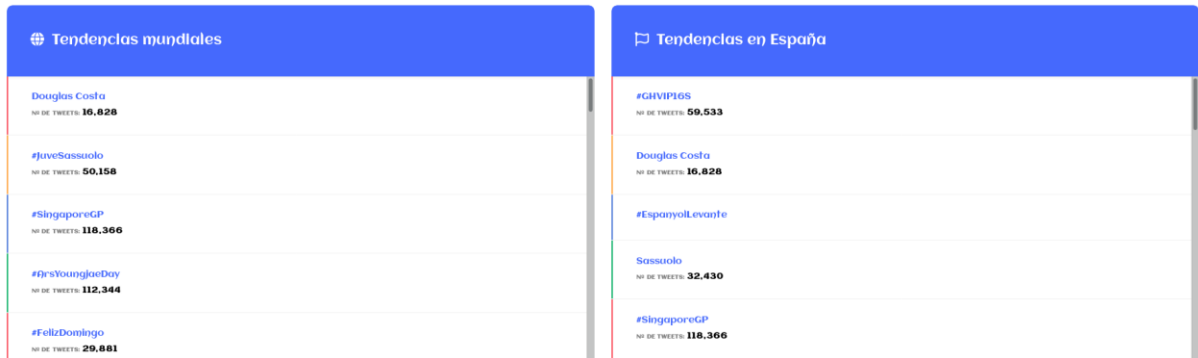


Figura 12: Tendencias.

3. Análisis

3.1	Requisitos.....	22
3.1.1	Requisitos funcionales	22
3.1.2	Requisitos no funcionales	23
3.1.3	Requisitos de información.....	24
3.2	Casos de uso	26

Sinopsis:

En este capítulo se reflejarán el análisis de requisitos del sistema. Estos requisitos proporcionan una descripción completa del comportamiento del sistema a desarrollar, e incluye un conjunto de casos de uso que especifica la interacción de los diferentes usuarios con el sistema.

3.1 Requisitos

3.1.1 Requisitos funcionales

Los requisitos funcionales que se presentan a continuación definen las funciones desarrolladas en el sistema por los diferentes componentes:

Identificador	Nombre	Descripción
RF01	Obtención de tuits	El sistema debe ser capaz de recuperar los tuits que contengan la palabra clave desde la API de Twitter
RF02	Obtención de los datos de los usuarios	El sistema debe ser capaz de recuperar los datos de los autores de los tuits
RF03	Análisis de sentimientos	El sistema analizará los sentimientos de los usuarios basándose en sus tuits.
RF04	Página de administración	El sistema contendrá una sección de administración en la que se podrá modificar el término de búsqueda y los diferentes datos almacenados en la base de datos.
RF05	Analizar el comportamiento de los usuarios	Se visualizará los datos de la actividad e impacto de los diferentes usuarios involucrados.
RF06	Relaciones entre usuarios	Se mostrará mediante técnicas de Visual Analytics las relaciones entre los usuarios en base a las siguientes acciones: respuestas, citas y menciones.
RF07	Análisis de propagación de tendencias	Se mostrarán las principales tendencias recogidas en los tuits (basadas en el tema buscado) y las tendencias generales a nivel mundial y nacional

RF08	Análisis de la difusión	Se recogerá la evolución en la participación de los usuarios (número de tuits enviados) así como el número total de participantes
------	-------------------------	---

3.1.2 Requisitos no funcionales

En este apartado se especifican las características que debe mantener el sistema:

Identificador	Nombre	Descripción
RNF01	Integridad	Los datos almacenados deberán ser fieles a los obtenidos de la red social
RNF02	Seguridad	Los datos de la aplicación solo podrán ser modificados por personas autorizadas para ello.
RNF03	Mantenibilidad	Código claro y comentado
RNF04	Capacidad de carga	El sistema debe ser capaz de gestionar el gran volumen de datos proporcionados por la API de Twitter.

3.1.3 Requisitos de información

El sistema deberá almacenar para su correcto funcionamiento la siguiente información:

Identificador	Nombre	Descripción
RI01	Tuits	<p>La aplicación debe mantener los datos recuperados sobre los mensajes de los usuarios (tuits).</p> <p>Los campos principales son:</p> <ul style="list-style-type: none">• User_id• Status_id• Created_at• Screen_name• Text• Is_retweet• Reply_to_status_id
RI02	Usuarios	<p>La aplicación debe mantener los datos recuperados sobre los usuarios/autores de los tuits.</p> <p>Los campos principales son:</p> <ul style="list-style-type: none">• User_id• Screen_name• Followers_count• Friends_count• Favourites_count
RI03	Palabra clave	<p>Se debe almacenar el término a buscar para la obtención de la información correspondiente.</p> <p>Los campos principales son:</p> <ul style="list-style-type: none">• Text• Description• NTuits

RI04	Histórico de búsquedas	<p>Se debe mantener un registro de todas las búsquedas realizadas.</p> <p>Los campos principales son:</p> <ul style="list-style-type: none"> • Search • Activa • Date • NTweets • Description • Complete
------	------------------------	--

3.2 Casos de uso

El diagrama siguiente representa las posibles interacciones de los diferentes usuarios con el sistema:

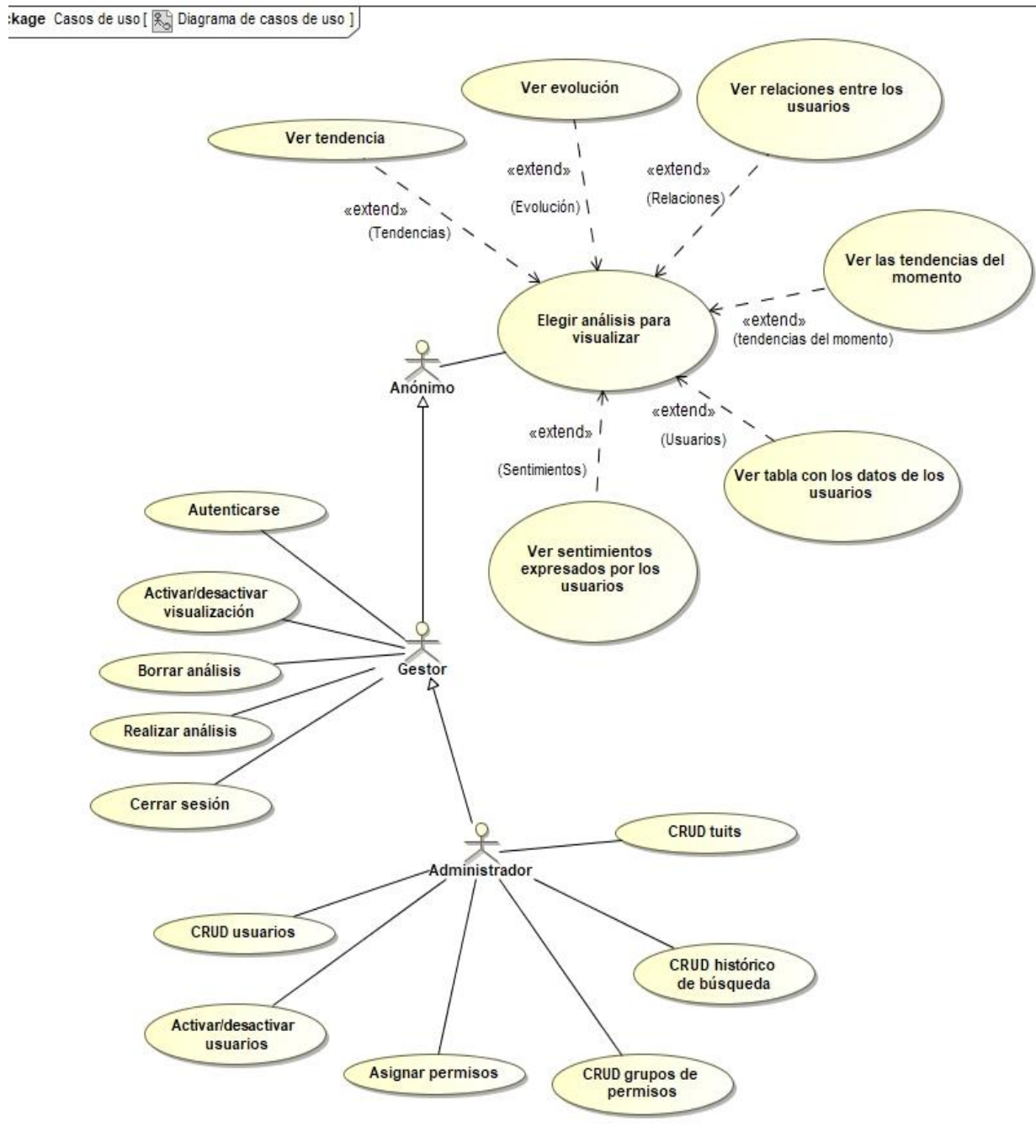


Figura 13: Diagrama de casos de uso

En las siguientes tablas se muestran una descripción detallada de cada una de las diferentes acciones que pueden realizar los diferentes actores.

Nombre	Visualizar análisis
Actor	Anónimo
Descripción	Un usuario anónimo podrá seleccionar los datos a visualizar de entre los diferentes análisis disponibles.

Nombre	Ver tendencia
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver las tendencias encontradas en los tuits obtenidos.

Nombre	Ver evolución
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver los datos sobre la evolución de los elementos analizados.

Nombre	Ver relación entre los usuarios
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver los diferentes grafos con las distintas relaciones entre usuarios.

Nombre	Ver las tendencias del momento
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver las tendencias mundiales y en España en el momento en el que se realizó el análisis

Nombre	Ver las tablas con los datos de los usuarios
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver los datos obtenidos y calculados de los usuarios participantes en la comunicación.

Nombre	Ver sentimientos expresados por los usuarios
Actor	Anónimo
Descripción	Un usuario anónimo podrá ver representados los sentimientos expresados en los tuits.

Nombre	Login
Actor	Gestor
Descripción	Un gestor tendrá la opción de autenticarse para acceder a las opciones propias de su perfil.

Nombre	Activar/desactivar visualización
Actor	Gestor
Descripción	Un gestor podrá seleccionar que análisis esta visible para los usuarios anónimos.

Nombre	Borrar análisis
Actor	Gestor
Descripción	Un gestor podrá borrar un análisis realizado y todos los elementos relacionados con él de forma permanente.

Nombre	Realizar análisis
Actor	Gestor
Descripción	Un gestor podrá definir los parámetros sobre los cuales realizar el análisis. Los parámetros son: <ul style="list-style-type: none"> • Término a buscar • Descripción • Numero de tuits

Nombre	Cerrar sesión
Actor	Gestor
Descripción	Un gestor podrá cerrar la sesión actual.

Nombre	CRUD usuarios
Actor	Administrador
Descripción	Un administrador podrá crear, ver, editar y borrar a los usuarios autenticados (gestores) de la aplicación.

Nombre	Activar/desactivar usuarios
Actor	Administrador
Descripción	Un administrador podrá desactivar la cuenta de usuarios autenticados (gestores) de la aplicación.

Nombre	CRUD tuits
Actor	Administrador
Descripción	Un administrador podrá crear, editar, ver y borrar los tuits.

Nombre	CRUD histórico de búsqueda
Actor	Administrador
Descripción	Un administrador podrá crear, editar, ver y borrar el histórico de búsqueda.

4. Diseño

4.1	Estructuras de datos.....	32
4.1.1	Modelo lógico.....	32
4.1.2	Modelo relacional.....	36
4.2	Estructura del proyecto.....	38
4.3	Metodología de desarrollo	41
4.4	Funcionamiento del sistema.....	42
4.4.1	Django	42
4.4.2	R	48

Sinopsis:

En este capítulo se expondrá el diseño de la aplicación mediante diferentes modelos y diagramas.

4.1 Estructuras de datos

La definición del modelo de datos es de fundamental importancia ya que es la base para el correcto funcionamiento del resto de la aplicación y un correcto diseño facilitará la explotación de los datos.

En esta sección se define las diferentes entidades involucradas y sus relaciones.

4.1.1 Modelo lógico

El modelo lógico es un diseño de alto nivel usado para definir las entidades, atributos y relaciones de los datos pertenecientes al sistema, siendo esta la base para la construcción de los diferentes sistemas de almacenamiento de datos que se incluyan en la aplicación.

Su objetivo es la definición del esquema conceptual del sistema independientemente del modo en el que se gestionen físicamente.

Es particularmente útil en el diseño realizado en este proyecto, debido a que en él se han utilizado diversas técnicas para el almacenamiento de los datos.

A continuación, se describe las entidades con sus atributos, así como sus relaciones, mediante el modelo lógico que se muestra en la siguiente figura de la siguiente página.

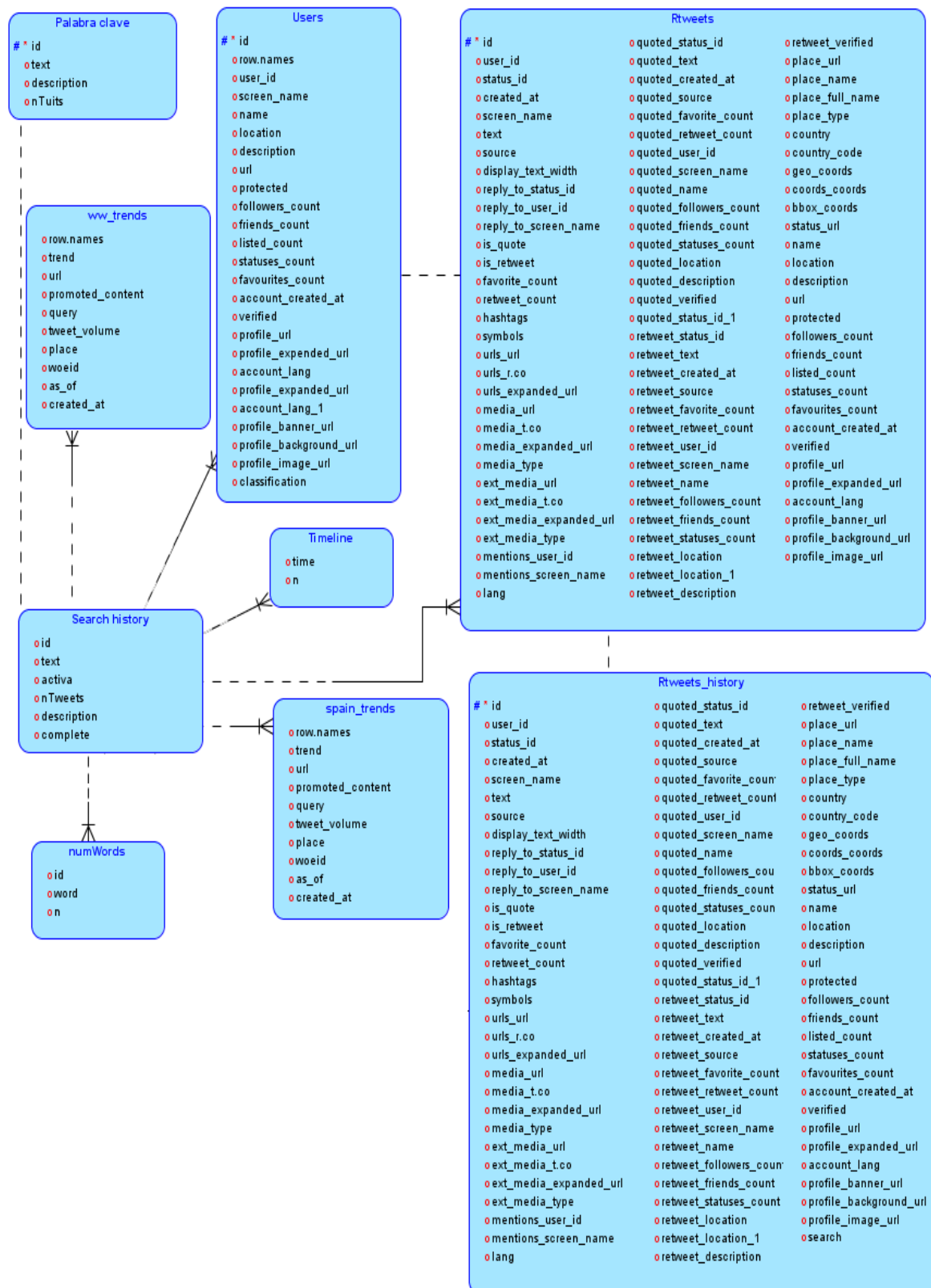


Figura 14: Modelo lógico de la base de datos

A continuación, se detallan las entidades mostradas en el modelo de la figura anterior:

- **Palabra clave:** Tabla que contiene el término a estudio, el cual será utilizado por la aplicación para la obtención de los datos a analizar.
- **Spain_trends:** Contiene tendencias del momento del estudio en España, aunque algunos campos significativos como el número de tuits (tweet_volume), no están disponibles en todas las tendencias.
- **Ww_trends:** Contiene tendencias del momento del estudio a nivel mundial, aunque con las mismas limitaciones que las de España, se mantienen en una entidad separadas por simplicidad a la hora de la obtención y procesamiento de los datos.
- **Users:** Contiene datos referentes a los autores de los tuits procedentes de la API de twitter para el análisis de su actividad e impacto.
- **Rtweets:** Tabla principal, contiene todos los datos de los tuits extraídos desde la API de twitter, teniendo un papel más relevante para el sistema los siguientes campos:
 - **User_id:** Identificador único del usuario en Twitter.
 - **Status_id:** Identificador único del tuit en Twitter.
 - **Created_at:** Fecha de creación del tuit.
 - **Screen_name:** Nombre de la cuenta del autor del tuit.
 - **Text:** Texto del tuit.
 - **Reply_to_user_id:** Identificador único del usuario destinatario de un tuit de respuesta.
 - **Is_retweet:** Identifica si es un retuit (reenvío de un tuit de otro usuario)
 - **Favourite_count:** Número de veces que el tuit ha sido marcado con “me gusta”.
 - **Retweet_count:** Número de veces que el tuit ha sido reenviado (“retuiteado”).
 - **Mentions_user_id:** Identificadores de los usuarios mencionados en el tuit.

- **Retweet_status_id:** Identificador del tuit original del que procede el retuit (reenvío).
- **Retweet_user_id:** Identificador del usuario creador del tuit original del que procede el retuit (reenvío).
- **Followers_count:** Número de “Seguidores” del usuario creador del tuit.
- **Friends_count:** Número de “Seguidores” del usuario creador del tuit.
- **Listed_count:** Número de listas públicas del usuario es miembro.
- **Classification:** Campo calculado, no procedente directamente de la API de twitter, especifica la clasificación del usuario en función de su actividad en el tema analizado.
- **Rtweets_history:** Histórico con todos los tweets obtenidos de los diferentes análisis, añadiendo un campo adicional para poder identificar el análisis de procedencia.
- **Search history:** Información acerca de todos los análisis realizados, es la entidad utilizada para agrupar toda la información relativa a un análisis.
 - **Text:** Término analizado.
 - **Activa:** Define si los resultados son visibles o no desde la aplicación.
 - **NTweets:** Número de tuits objetivo para realizar el análisis.
 - **Description:** Descripción de la búsqueda. Utilizado para añadir información relativa al análisis.
 - **Complete:** Define si el análisis ha sido completado.
- **Timeline:** Almacena la evolución en cuanto a la participación de los usuarios.
- **NumWords:** Número de apariciones de palabras en los tuits.
- **Django_session:** Datos sobre las sesiones activas en la aplicación.
- **Django_migrations:** Información sobre las migraciones realizadas en la base de datos de la aplicación.
- **Django_content_type:** Información de todos los modelos instalados en la aplicación.
- **Auth group:** Grupos de usuarios.
- **Auth user:** Usuarios de la aplicación.
- **Auth_permission:** Permisos de la aplicación.
- **Django_admin_log:** Información para auditorias de la aplicación.

4.1.2 Modelo relacional

Para el almacenamiento y explotación de las entidades anteriormente expuestas se han utilizado dos estructuras diferentes, dependiendo del uso requerido de los datos que contienen.

Las entidades esenciales para el funcionamiento de la aplicación y las relativas a datos históricos se han almacenado usando una base de datos relacional, en concreto una base de datos bajo el sistema gestor de bases de datos postgresQL.

Las entidades relativas a datos exclusivos de resultados de análisis se han almacenado en archivos con formato CSV (comma-separated-values), en carpetas relacionadas con el análisis al que pertenecen, con el fin de mantener una estructura clara y de fácil mantenimiento.

En esta sección definiremos el contenido de cada una de las tablas almacenadas en la base de datos, especificando cada campo, tipo y función.

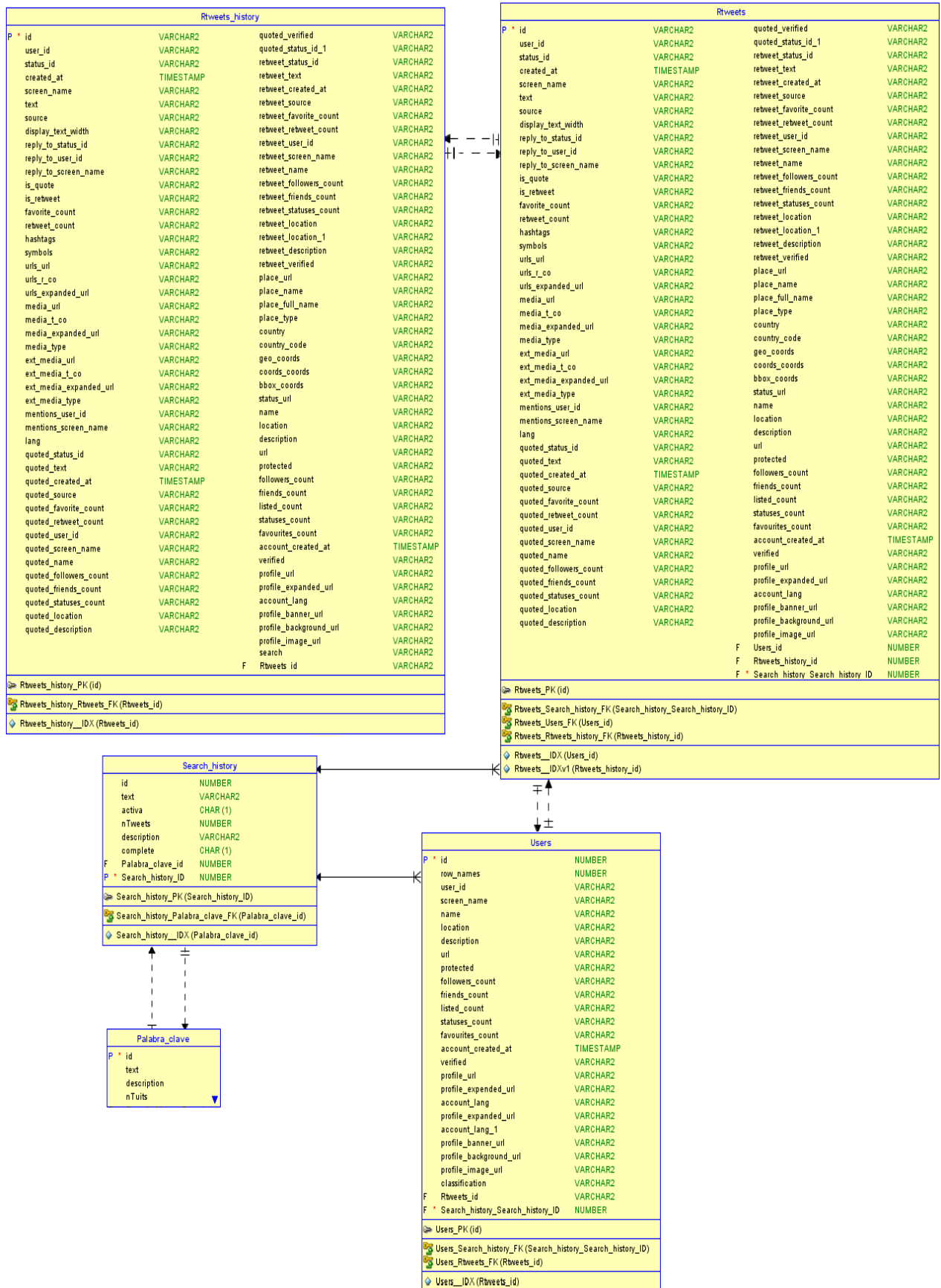


Figura 15: Modelo relacional de la base de datos (1)

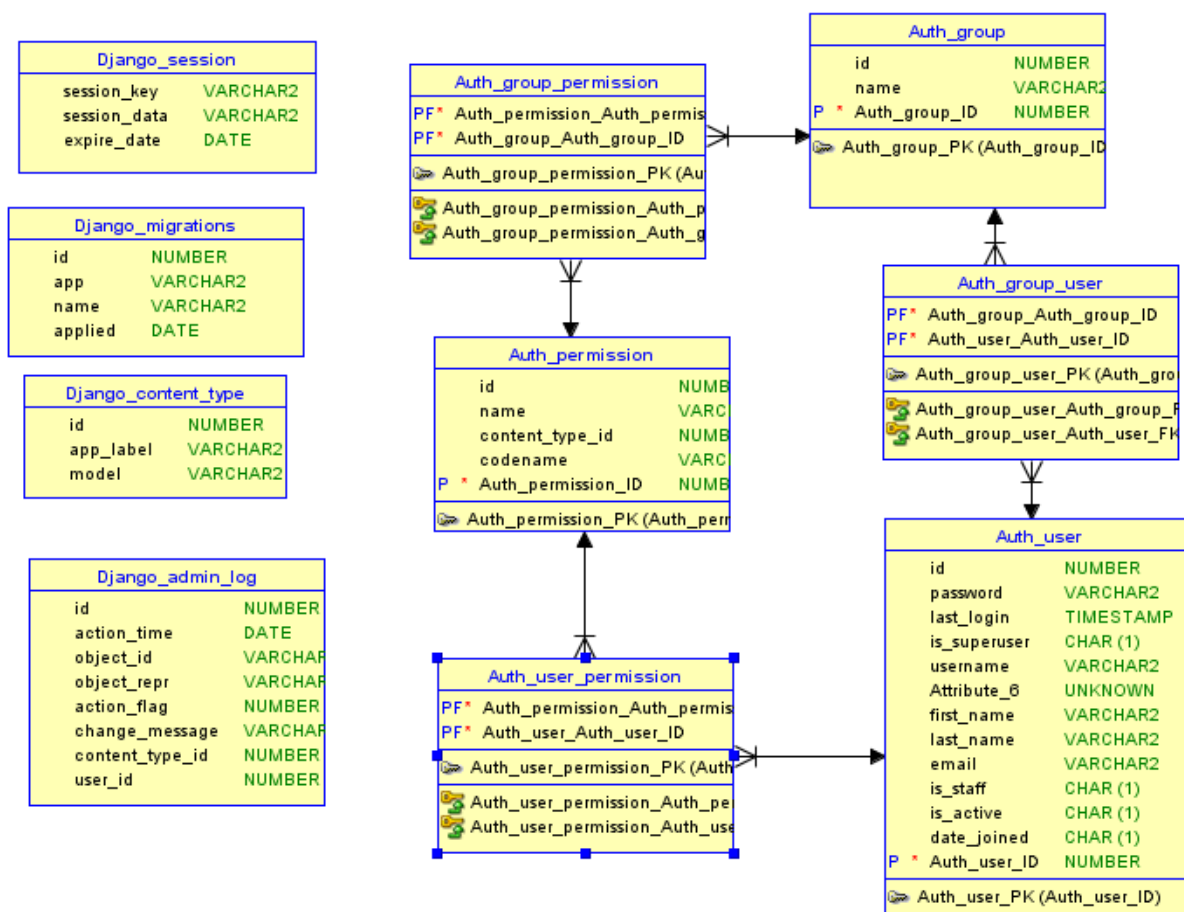


Figura 16: Modelo relacional de la base de datos (2)

En este caso la mayoría de los campos son de texto, ya que los análisis son mayormente sobre este tipo de dato y simplifica su recuperación y almacenamiento con la librería utilizada.

4.2 Estructura del proyecto

La estructura del proyecto se basa en la estructura básica proporcionada por el framework de desarrollo utilizado, a la que se le añaden rutas específicas para el almacenamiento de los diferentes resultados de los análisis realizados.

Estas rutas concretas son especificadas en el fichero de configuración del framework (settings.py), y se envían como parámetros a los diferentes scripts de R. De esta forma para cada análisis se crearán carpetas en las rutas especificadas con un nombre

identificativo compuesto por el término buscado y la fecha con el fin de tener los datos organizados y poder recuperarlos fácilmente.

Debido a la naturaleza de esos resultados hay tres rutas específicas para su almacenamiento:

- Las aplicaciones creadas desde R para la visualización de datos se almacenan junto a los templates de la propia aplicación (templates).
- El resto de los datos se almacenan en una ruta de archivos estáticos (static), en una carpeta nombrada R en la base del proyecto. Ésta, a su vez se divide en dos partes:
 - Los archivos con formato CSV se almacenan en una carpeta de igual nombre.
 - Las imágenes se almacenan en una carpeta de nombre “images”.

Los diferentes scripts de R, necesarios para el procesamiento están almacenados en la carpeta de contenido estático en la base del proyecto (static).

Se añaden dos ficheros en formato CSV necesarios para el procesado en la carpeta CSV en la ruta de ficheros estáticos. Dichos ficheros son `sdal.csv` y `stopwords_es.csv`, usados para el análisis de sentimientos.

En la página siguiente se muestran las diferentes rutas mencionadas sobre la estructura básica del proyecto:

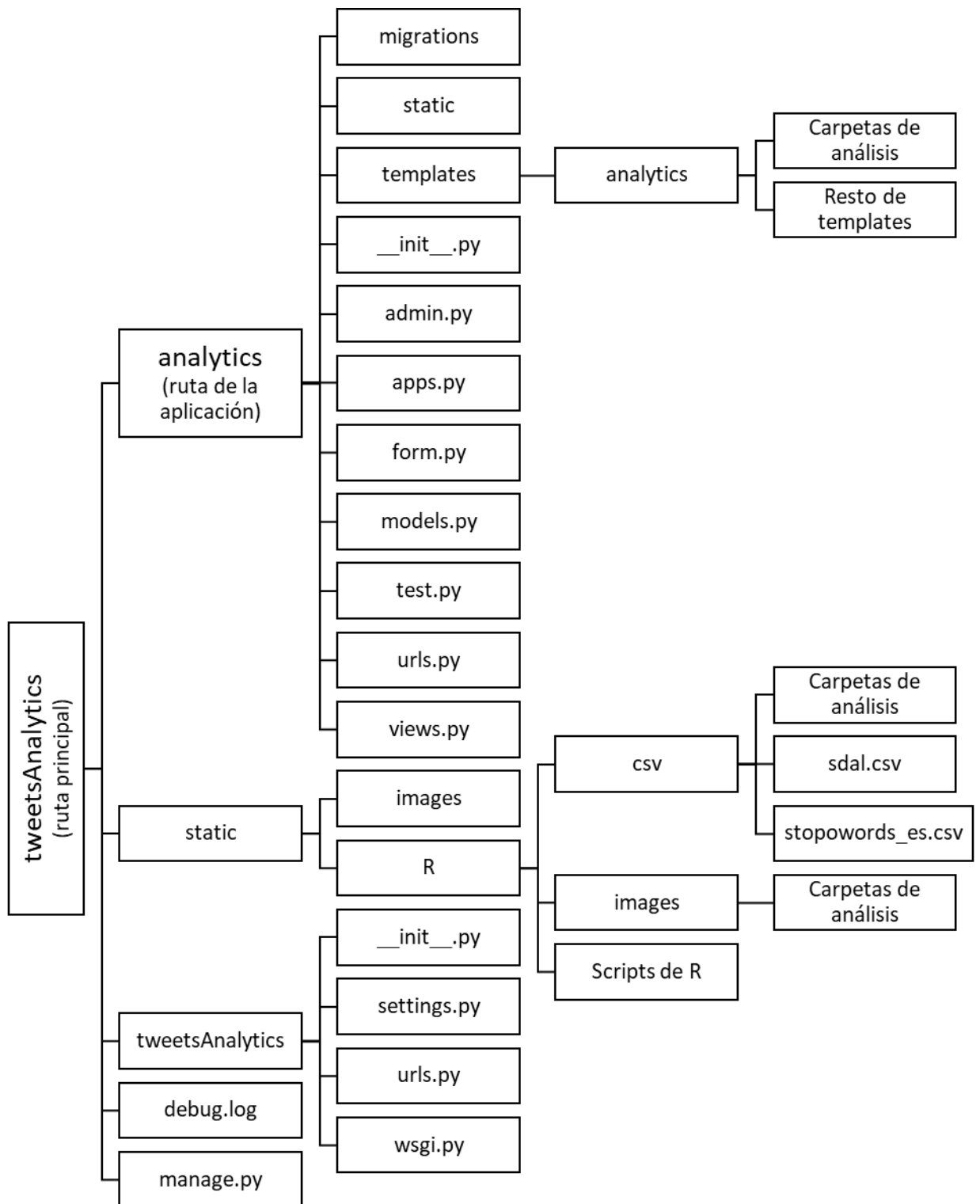


Figura 17: Estructura del sistema

4.3 Metodología de desarrollo

La metodología elegida para la elaboración del proyecto ha sido el modelo de desarrollo Scrum.

Scrum es una metodología ágil cuyo objetivo es la de planificar proyectos flexibles, con una incertidumbre elevada.

La planificación se basa en Sprint o iteración, términos que hacen referencia a un periodo de tiempo de un par de semanas. Al final de cada periodo se revisa el trabajo realizado el periodo anterior y en función de eso, se priorizan y planifican las actividades para el sprint siguiente.

Entre las principales ventajas de esta metodología encontramos la flexibilidad y adaptación a los cambios, alineamiento entre el cliente y el equipo de desarrollo al haber un representante de este (Project Owner) y la reducción del riesgo al desarrollar primero las funcionalidades con mayor interés.

Para la definición de los diferentes sprint se han tomado como referencia las fases de trabajo definidas en el anteproyecto, las cuales se numeran a continuación:

1. Análisis de requisitos y modelado inicial de la aplicación.
2. Extracción y almacenamiento de los datos proporcionados por la API. Se definirá un modelo de persistencia adecuado que permita almacenar la información para su posterior tratamiento y se implementarán los procedimientos necesarios para la obtención de dichos datos.
3. Análisis de propagaciones. Se desarrollarán procedimientos para identificar la fuente de la información y la propagación de ésta.
4. Análisis de comportamiento de los usuarios. Se implementarán métodos para identificar la actividad y el impacto de los usuarios según su uso de la red social.
5. Búsqueda de relaciones. Se implementarán procedimientos que identifiquen las relaciones entre usuarios según sus acciones en la red social y así definir segmentos de usuarios.
6. Análisis de la propagación de tendencias. Se implementarán métodos que distingan los temas de mayor interés entre los usuarios y cómo se propagan a través de la red social.

4.4 Funcionamiento del sistema

El sistema desarrollado consta de dos partes diferenciadas, una aplicación web cuyo objetivo es permitir la interacción del usuario con la aplicación y los diferentes scripts escritos en el lenguaje de programación R, los cuales realizan el análisis mediante técnicas de Visual Analytics.

Este proceso comporta diversas fases, las cuales se descomponen en las siguientes:

- Data extraction: Fase en la que se recuperan los tweets a través de la API de Twitter.
- Data wrangling: En esta fase los datos extraídos son filtrados y se le aplica una conversión de formato, concretamente se eliminan algunos símbolos espúreos y se modifica el tipo de dato.
- Data analysis: Fase en la que se procesan los datos ya filtrados en los diferentes scripts.
- Data Visualization: Fase final en la que se muestran mediante herramientas gráficas los datos analizados al usuario.

De esta forma se procede a explicar ambas partes de forma independiente, con el fin de simplificar su comprensión.

4.4.1 Django

El sistema desarrollado consiste en una aplicación web desarrollada bajo el framework Django, cuyo objetivo principal es la de proveer acceso a las diferentes funcionalidades del sistema, tanto a los resultados de los análisis realizados como a la gestión de los datos obtenidos.

Esta funcionalidad se consigue siguiendo la estructura que define el propio framework, en la que el procesamiento de las diferentes peticiones HTTP se ejecuta en las diferentes vistas.

Seguidamente se procede a exponer las vistas que por su importancia destacan en el sistema, dejando de lado funcionalidades propias del framework como la vinculación entre las llamadas y las vistas (enrutado) o el sistema de plantillas:

Analyze: es la función que permite el análisis de los términos seleccionados mediante hebras, con el fin de ser una acción en segundo plano, es decir, que sea transparente al usuario, para que pueda continuar con el uso de la aplicación mientras se realizan los análisis.

```
def analyze(request):
    if not request.user.is_authenticated:
        return JsonResponse({"result": "error"})

    hilo1 = threading.Thread(target=getTweets, args={'request': request}, daemon=True)
    hilo1.start()
    return JsonResponse({"result": "ok"})
```

GetTweets: Es la función que invoca el script de R encargado de la recuperación de los tweets y crea las carpetas necesarias para el almacenamiento de los resultados de los posteriores análisis.

Esta función se puede dividir en tres fases:

- La espera activa de las hebras.
 - Los procesos de obtención de los tweets se han realizado de forma secuencial por limitaciones de la API, ya que puede devolver errores al recibir demasiadas llamadas en breves periodos de tiempo.

```
"""espera activa de las hebras"""
BusquedasActivas = Search_history.objects.filter(complete=False).count()
while BusquedasActivas > 1:
    BusquedasActivas = Search_history.objects.filter(complete=False).count()
    sleep(60)
```

- La invocación del script de obtención de datos y la creación de las carpetas para almacenar los resultados de los análisis posteriores.

```

args = [key.text, str(key.nTuits)]
"""ejecuta comando y recupera salida"""
cmd = [command, path2script] + args
x = subprocess.check_output(cmd, universal_newlines=True)

"""ruta carpeta para guardado de archivos"""
carpeta = busquedaH.search + busquedaH.date.strftime("%Y-%m-%d--%H-%M-%S")

"""crea rutas para guardar los cálculos si no existen"""
if not os.path.exists(os.path.join(settings.MEDIA_ROOT + settings.STATIC_R+'csv/', carpeta)):
    os.mkdir(os.path.join(settings.MEDIA_ROOT + settings.STATIC_R+'csv/', carpeta))

if not os.path.exists(os.path.join(settings.MEDIA_ROOT + settings.STATIC_IMAGE_R, carpeta)):
    os.mkdir(os.path.join(settings.MEDIA_ROOT + settings.STATIC_IMAGE_R, carpeta))

if not os.path.exists(os.path.join(settings.ANALYTICS_APP, carpeta)):
    os.mkdir(os.path.join(settings.ANALYTICS_APP, carpeta))

```

- La ejecución de los análisis mediante llamadas a los diferentes scripts y el control de posibles errores en su ejecución.

```

try:
    userClasification(request, busquedaH)
except:
    logger.error('error en script userClasification.R')
    shutil.rmtree(os.path.join(settings.MEDIA_ROOT + settings.STATIC_R, "csv/" + carpeta))
    shutil.rmtree(os.path.join(settings.MEDIA_ROOT + settings.STATIC_IMAGE_R, carpeta))
    shutil.rmtree(os.path.join(settings.ANALYTICS_APP) + carpeta)
    busquedaH.delete()

```

- Se muestra la llamada a un análisis a modo de ejemplo debido a la similitud con el resto.

Métodos de invocación de scripts:

Los métodos utilizados para la ejecución de los diferentes scripts de R, son similares por lo que se mostrará uno general a modo de ejemplo.

Los métodos relativos a la ejecución de los diferentes scripts se encargan de:

- Obtener las rutas en las que se deben almacenar los resultados del análisis a realizar, con el fin de proporcionar esa información al script en forma de argumentos del comando.
- Ejecutar dicho comando y recuperar la salida para confirmar la correcta ejecución de éste.
- Devolver una respuesta en formato Json.

A continuación se muestra un ejemplo de esta funcionalidad en el que se pueden observar los procesos anteriormente expuestos:

ObtenerTendencias: Este método ejecuta el script relativo a la obtención de tendencias.

```
def obtenerTendencias(busqueda=''):
    command = 'Rscript'
    pclave = Search_history.objects.latest('date') if busqueda == '' else busqueda

    """ruta al script"""
    path2script = os.path.join(settings.MEDIA_ROOT + settings.STATIC_R, "tendencias.R")

    """ruta carpeta para guardado de archivos"""
    carpeta = pclave.search + pclave.date.strftime("%Y-%m-%d--%H-%M-%S")

    """ruta para guardar la imagen"""
    args = [os.path.join(settings.MEDIA_ROOT + settings.STATIC_R, "csv/" + carpeta + "/spainTrends.csv"),
            os.path.join(settings.MEDIA_ROOT + settings.STATIC_R, "csv/" + carpeta + "/wwTrends.csv")]

    """ejecuta comando y recupera salida"""
    cmd = [command, path2script] + args
    x = subprocess.check_output(cmd, universal_newlines=True)
    return JsonResponse({"result": "ok"})
```

Métodos de exposición de resultados:

Estas funciones se encargan de recuperar los datos almacenarlos y devolver una representación adecuada de estos mediante el uso de templates (plantillas previamente definidas).

Estas funciones reciben como parámetro opcional la búsqueda o análisis del cual deben mostrar los datos, este parámetro es opcional debido a que en su ausencia se mostrará el último análisis completo y activo.

Los datos son recuperados de las carpetas anteriormente definidas, las cuales se crean con un formato específico, explicado en la sección de arquitectura del sistema, que permite la fácil recuperación de los datos. Además, en algunas de estas vistas se realiza un pequeño procesamiento para recuperar la información almacenada en formato CSV.

Seguidamente se añade un ejemplo de este tipo de métodos, en el que se observa fácilmente las características mencionadas:

```

"""vista de tendencias del termino buscado"""
def tendencias(request, busqueda=''):
    pclave = Search_history.objects.get(id=busqueda) if busqueda != '' else Search_history.objects.filter(activa=True, complete=True).latest('date')
    """ruta carpeta para guardado de archivos"""
    carpeta = pclave.search + pclave.date.strftime("%Y-%m-%d--%H-%M-%S")

    """importo csv del wordCount del script de R"""
    ifile = open(os.path.join(settings.MEDIA_ROOT + settings.STATIC_R, "csv/" + carpeta + "/numWords.csv"), "r")
    reader = csv.reader(ifile)

    rownum = 0
    words = []
    nWords = []
    for row in reader:
        if rownum == 0:
            header = row
        else:
            colnum = 0
            for col in row:
                if header[colnum] == "word":
                    words.append(col)
                elif header[colnum] == "n":
                    nWords.append(int(col))
                colnum += 1
            rownum += 1

    ifile.close()

    return render(request, 'analytics/tendencias.html',
        {
            "nube": settings.STATIC_IMAGE_R+carpeta+"/wordsCloud.png",
            "totalWords": sum(nWords),
            "words": json.dumps(list(words[:8]), cls=DjangoJSONEncoder),
            "nWords": json.dumps(list(nWords[:8]), cls=DjangoJSONEncoder),
            "busqueda": pclave,
            "clave": PalabraClave.objects.last()
        })

```

Por último, están las funciones pertenecientes a la autenticación del usuario y de gestión del histórico de búsqueda, ambas relacionadas con la funcionalidad relativa al usuario con el rol de gestor.

En lo concerniente a las funciones relativas a la autenticación, el framework Django provee de funciones específicas para gestionar tanto la autenticación como la autorización, siendo muy sencillo el hecho de implementar el control de acceso en la aplicación.

Aquí podemos observar el método utilizado para la autenticación en la aplicación, en el cual se utilizan las funciones propias del framework: `authenticate`, para la verificación de las credenciales del usuario y `login`, para crear la sesión de éste.

```
def loginCustom(request):
    if request.method == 'POST':
        user = authenticate(username=request.POST['username'], password=request.POST['password'])
        if user is not None:
            login(request, user)
            """messages.success(request, 'Acceso correcto.')"
            return redirect('gestor')
        else:
            messages.error(request, 'Acceso incorrecto.')
            next = request.POST.get('next', '/')
            return HttpResponseRedirect(next)
```

De la misma manera provee una función para finalizar la sesión del usuario:

```
def logoutCustom(request):
    logout(request)
    return redirect('index')
```

Por último, se expondrá un ejemplo de una función relativa a la gestión del histórico de búsqueda, en la cual se añade un control de acceso, con el fin de que solo un usuario autenticado, pueda utilizar dichas funciones.

```
def activar(request):
    if not request.user.is_authenticated:
        messages.error(request, 'Error de acceso.')
        return index(request)
    id = request.GET['id']
    historico = Search_history.objects.get(id=id)
    historico.activa = not historico.activa
    historico.save()
    return HttpResponseRedirect('gestor')
```

Además de la funcionalidad previamente explicada y accesible desde la aplicación principal, Django provee de una interfaz de administración automática, en la cual se han registrado los diversos modelos utilizados en la aplicación.

Esta herramienta de administración permite que un usuario con el rol de administrador, gestione las diferentes entidades de la aplicación.

En concreto desde la herramienta de administración se podrán gestionar las tablas del histórico de búsqueda, la palabra clave, los tweets obtenidos en el último análisis. Además de la gestión de usuarios y contraseñas, grupos y permisos.

4.4.2 R

Los procesos de extracción, procesamiento y análisis de datos se han realizado íntegramente en R, con el propósito de aprovechar sus características, las cuales permiten el trabajo con grandes volúmenes de datos.

La aplicación se ha dividido en ocho scripts que realizan los diferentes análisis. Estos scripts se han diseñado con el objetivo reducir lo máximo posible el acoplamiento entre ellos, a pesar de lo cual, sí existen dependencias entre algunos de ellos, principalmente con el script que realiza la carga de tweets y entre los relativos al cálculo de comunidades.

Los scripts de la aplicación son los siguientes (no se transcribe su código por la extensión):

- **CargaTweets1.R:** El script principal, es el que realiza la carga de tweets, el cual es necesariamente el primero en ejecutarse.
 - Su funcionalidad se puede dividir en varias fases:
 - La recuperación de los tweets.
 - Para ello se debe crear un token, con los datos que proporciona Twitter para el uso de su API. Una vez creado el token se almacena como variable de entorno, por lo que no es necesario repetir el procedimiento.
 - El procesamiento de los datos obtenidos, que consisten principalmente en el cambio de tipos para el almacenamiento en la base de datos.
 - El almacenamiento de los datos, añadiendo información adicional para posteriores análisis.
- **GrafoRelaciones.R:** En este script se realizan los siguientes análisis:
 - Creación de un gráfico de análisis de las relaciones de usuarios en base a menciones, que se crea como una aplicación independiente a fin de que mantenga la interactividad.

- Creación de un gráfico de análisis de sentimientos, aprovechando los cálculos previamente realizados.
 - Creación de la gráfica estadística que representa los sentimientos expresados por los usuarios.
- **GrafoRelaciones2.R:** Script que realiza el análisis de las relaciones de usuarios en base a respuestas y la creación del gráfico correspondiente.
 - **GrafoRelaciones3.R:** Script que analiza las relaciones entre los diferentes usuarios en base a retweets, entendiendo como retweet la acción de compartir publicaciones de otros usuarios.
 - **Tendencias.R:** Recupera las tendencias a nivel mundial y nacional.
 - **Timeline.R:** Representa la difusión del término analizado.
 - **UserClasification.R:** Realiza la clasificación de los diferentes roles ejercidos por los usuarios en base a su actividad. Dicha clasificación se basa en la tesis de D^a. Mariluz Congosto, doctora en Telemática por la Universidad Carlos III.
 - **WordCLoud2.R:** Analiza la frecuencia de aparición de palabras y crea el grafico oportuno.

5. Tecnologías

5.1	Tecnologías Web.....	52
5.1.1	HTML.....	52
5.1.2	Javascript.....	52
5.1.3	CSS	53
5.2	Lenguajes de programación / Script.....	53
5.2.1	Python.....	53
5.2.2	R	54
5.3	Persistencia.....	56
5.3.1	PostgreSQL	56
5.4	Framework	57
5.4.1	Django	57

Sinopsis:

En este capítulo se detallarán las diferentes tecnologías usadas en el desarrollo de este proyecto, así como sus principales características.

5.1 Tecnologías Web

5.1.1 HTML

HTML, (HyperText Markup Language), es el lenguaje básico para la creación de cualquier página web.

Es utilizado para definir la estructura de la página web y los distintos elementos que la componen.

Es un estándar del World Wide Web Consortium (W3C), organización encargada de la estandarización de tecnologías utilizadas en internet.

Su principal importancia radica en que es el estándar que se ha impuesto para la visualización de páginas web y es un estándar soportado por todos los navegadores.

Se ha utilizado la versión actual, conocida como HTML5 o HTML versión 5, publicada en octubre del 2014.

Se ha desarrollado partiendo de la plantilla web CoolAdmin, disponible bajo licencia MIT (Colorlib).

5.1.2 Javascript

JavaScript (JS) es un lenguaje de programación imperativo e interpretado que permite dotar de dinamismo a las páginas web mediante scripts que se ejecutan en el navegador.

También se puede utilizar en entornos de programación multiplataforma que se ejecutan del lado del servidor.

El estándar actual ECMAScript 5.1 es soportado por la mayoría de los navegadores.

5.1.3 CSS

CSS (Cascading Style Sheets), es un lenguaje de diseño que permite definir el estilo que tendrán los distintos componentes de la página web.

Es el lenguaje comúnmente utilizado para definir la apariencia de los elementos que componen la web.

El uso de este lenguaje permite separar la estructura de la página de la presentación con el objetivo de proveer mayor flexibilidad y control a las características visuales.

La especificación este lenguaje viene dada por el World Wide WEB Consortium (W3C).

En este proyecto se ha utilizado la versión CSS 3 debido a que es la última versión de este lenguaje.

5.2 Lenguajes de programación / Script

5.2.1 Python

Python es un lenguaje interpretado perteneciente al paradigma imperativo, orientado a objetos.

Es un lenguaje multiplataforma, con licencia de código abierto lo que hace pueda ser utilizado en muchos escenarios diferentes.

Uno de sus principios básicos es la claridad y sencillez de su sintaxis lo que reduce su complejidad tanto para el desarrollo como para el mantenimiento, lo que lo hace interesante para el desarrollo rápido de aplicaciones.

Además, cuenta con una gran cantidad de librerías de libre distribución que lo dota de gran funcionalidad y facilitan la integración con otros lenguajes. Muchas de estas librerías están centradas en el trabajo con grandes volúmenes de datos lo que lo hace ser elegido para proyectos de Big Data.

Para el desarrollo del proyecto se ha utilizado la versión de Python 3.6.6.

5.2.2 R

R es un lenguaje para computación y gráficos estadísticos que toma como base gran parte del código escrito para el lenguaje de programación S añadiendo diferencias notables.

R fue creado en 1992 en Nueva Zelanda, por Robert Gentleman y Ross Ihaka del Departamento de Estadística de la Universidad de Auckland y publicado por primera vez en 1993. (Santana, et.al., 2014)

Una de las características destacables de R es la gran cantidad de librerías para la creación de gráficos totalmente parametrizables, que permiten al usuario definir las diferentes opciones de diseño posibles.

R dispone de gran variedad de librerías con funciones estadísticas y gráficos, a lo que añade el ser un lenguaje de código abierto fácilmente extensible.

R está disponible como Software Libre con licencia GNU GPL (General public license) que garantiza a los usuarios la libertad de usar, modificar y compartir el software.

En este proyecto se ha aprovechado la extensibilidad de este lenguaje mediante el uso de diferentes librerías tanto estadísticas como gráficos.

A continuación se mencionarán las librerías utilizadas en los diferentes procesos.

- Rtweet: librería utilizada para la obtención de datos procedentes de la API de Twitter.
- Httpuv: Librería para realizar manejar llamadas HTTP y Websocket.
- Dplyr: paquete que proporciona una gramática para la manipulación de dataframes, proporcionando mayor nivel de abstracción.
- RPostgreSQL: paquete que proporciona una interfaz con la base de datos (DBI), permitiendo el acceso a bases de datos postgresQL desde R.
- Igraph: Librería para el trabajo con grafos y el análisis de redes.
- Magrittr: librería que proporciona una sintaxis más sencilla, con el fin de simplificar el código y aumentar su legibilidad y mantenibilidad.
- visNetwork: paquete para la visualización de grafos. A su vez esta librería utiliza la librería de Javascript, vis.js.
- DiagrammeR: paquete para la generación y manipulación de grafos.

- `data.table`: librería que provee una sintaxis sencilla para la manipulación de datos.
- `htmlwidgets`: librería que permite la creación de pequeñas aplicaciones para la visualización de datos basadas en javascript.
- `tidyverse`: colección de paquetes diseñados para la manipulación y análisis de datos.
- `tidytext`: librería con funciones especializadas en minería de datos.
- `stringr`: librería para la manipulación de cadenas de texto.
- `ggrepel`: librería que permite añadir y manipular etiquetas en los gráficos.
- `scales`: paquete que permite ajustar la escala en la representación de los gráficos y la conversión de datos a valores porcentuales.
- `stopwords`: librería que provee de listas de palabras vacías en diferentes lenguajes. Las palabras vacías o stopwords son palabras que no aportan ningún significado para el análisis del texto, como artículos o pronombres.

Para el desarrollo de los diferentes scripts se ha utilizado el entorno de desarrollo integrado (IDE) especialmente diseñado para este lenguaje, RStudio, en su versión de código abierto.

Este entorno de desarrollo incluye una consola, un editor de texto y herramientas para la depuración, gestión del espacio de trabajo y la representación de elementos gráficos.

Al ser un editor desarrollado para este lenguaje incluye características que facilitan el desarrollo como el resaltado de sintaxis y finalización de código. Además, permite ejecutar el código directamente desde el editor e incorpora ayuda y documentación sobre las diferentes funciones del lenguaje.

5.3 Persistencia

5.3.1 PostgreSQL

PostgreSQL es un sistema de base de datos creado por la Universidad de California con más de 30 años de actividad.

Este sistema consiste en un gestor de bases de datos relacional abierto que utiliza el lenguaje SQL, aunque lo amplía con diversas funciones propias que mejoran las características que proporciona.

Este sistema gestor de bases de datos provee de diversas funcionalidades con el objetivo de ayudar a los desarrolladores a crear aplicaciones y a los administradores para proteger el sistema. Además, al ser un sistema fácilmente extensible permite definir sus propios tipos de datos o incluir código de diferentes lenguajes de programación.

PostgreSQL posee muchas características que lo dotan a de gran popularidad entre las cuales caben destacar la arquitectura, extensibilidad, integridad de los datos, confiabilidad y la gran comunidad de código abierto que le da soporte. Otras de las características importantes es la posibilidad de ejecutarlo en los principales sistemas operativos y los complementos con gran funcionalidad como el extensor de bases de datos geoespaciales PostGIS.

PostgreSQL no cumple totalmente con el estándar SQL, pero muchas de sus funciones son compatibles con este estándar, aunque algunas con una sintaxis diferente.

Además, es un sistema altamente escalable que puede soportar una gran cantidad de usuarios, por lo que es utilizado en muchos sistemas de clústeres en entornos de producción con cantidades masivas de información.

Este sistema de código abierto es desarrollado por la comunidad conocida como PGDG (PostgreSQL Global Development Group).

Para el desarrollo del sistema este sistema gestor de bases de datos PostgreSQL en su versión 10.5.

5.4 Framework

5.4.1 Django

Django es un framework diseñado para permitir el desarrollo de aplicaciones de forma rápida, segura y mantenible.

Es gratuito y de código abierto, con una amplia documentación y una comunidad activa y creciente.

Django fue diseñado para permitir la reutilización de código y patrones de diseño comunes por lo que se desarrolló como un framework genérico conocido como proyecto Django en julio de 2005.

Actualmente es un proyecto colaborativo con gran vitalidad, con miles de usuarios y contribuidores que se ha convertido en un framework versátil, útil para desarrollar cualquier sitio web.

En una arquitectura clásica de un sitio web, la aplicación espera peticiones HTTP del navegador web u otro cliente, cuando recibe la llamada ejecuta las funciones oportunas con los datos obtenidos en la petición y devuelve una respuesta al navegador, normalmente creando una página web de forma dinámica para representar los datos, basándose en plantillas HTML.

Las aplicaciones de Django separan estos pasos en diferentes ficheros para aislar cada funcionalidad, con la idea de mejorar la reusabilidad y facilitar el mantenimiento del código.

En la estructura básica de una aplicación en Django encontramos cuatro ficheros principales.

Urls.py, El fichero usado para vincular las URL de las diferentes llamadas a las funciones que las gestionan, es decir, reenvía las peticiones HTTP a la vista. Esta vinculación también puede recuperar parámetros de las URLs mediante expresiones regulares y etiquetas permitiendo enviarlos como parámetros en la vista.

Views.py, contiene todas las funciones (vistas) que procesan las diferentes peticiones recibidas por la aplicación y devuelven las respuestas oportunas. Estas vistas acceden

a los datos de la aplicación por medio de modelos (models) y devuelven los resultados mediante un sistema de plantillas (templates).

Models.py, Contiene los diferentes modelos, que son objetos definidos en Python, que definen la estructura de datos y proporciona los mecanismos para el tratamiento de estos.

Templates, un template es una plantilla que define la estructura de una página HTML mediante marcadores que son interpretados de forma dinámica para crear la página web.

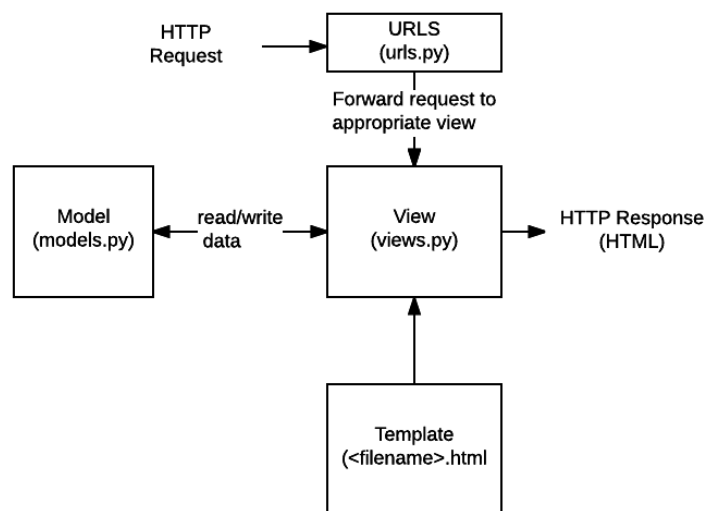


Figura 18: Esquema de funcionamiento del framework Django. (Mozilla)

Django también proporciona otras funcionalidades destacables como son la gestión de formularios, autenticación y permisos de usuarios, sitio de administración por defecto, y serialización de datos.

Para el desarrollo del sistema se ha utilizado la versión 2.1 de este framework.

5.5 Integración

Para la realización de este proyecto se han integrado dos lenguajes diferentes con el fin de aprovechar las características propias de cada uno de ellos. Como consecuencia se han tenido que llevar a cabo diversas técnicas para la comunicación entre estos lenguajes.

Para comenzar se ha utilizado R como lenguaje para la recuperación, manipulación y almacenamiento inicial de los datos procedentes de la API de Twitter, para lo cual se ha utilizado la librería Rtweet que permite la comunicación con dicha API.

Para la comunicación de los datos obtenidos se utilizó una base de datos común, la cual será utilizada desde los diversos scripts de R y por la propia aplicación web escrita en Python. Para ello la información inicial es almacenada desde R, usando la librería RPostgreSQL, que provee de una interfaz para la comunicación con la base de datos PostgreSQL.

Antes de almacenar los datos obtenidos desde la API se realiza un preprocesado de la información centrado en el cambio de tipos propios de R con el fin de que se almacenen correctamente y sean accesibles desde ambas aplicaciones.

El resto de los scripts de R accederán directamente a la base de datos recuperando los datos correspondientes para realizar los distintos análisis, debido a que, aunque actualmente se realizan todos los análisis de forma secuencial, están definidos en scripts independientes con el propósito de que se puedan realizar de forma independiente.

Los resultados obtenidos de los diferentes scripts en R se almacenan en rutas concretas de la aplicación de tres formas diferentes dependiendo de la naturaleza de estos. Estos resultados se almacenan en ficheros con formato CSV, en imágenes con formato .PNG o pequeñas aplicaciones para la visualización mediante Javascript.

Para la recuperación y gestión de los datos iniciales en el lenguaje de programación Python, se definen los modelos propios del framework de desarrollo utilizado (Django), y se configura el acceso a la base de datos PostgreSQL.

Para el uso de una base de datos PostgreSQL desde Python es necesario instalar el paquete Psycopg2.

En cuanto a los datos producidos posteriormente por los diferentes scripts de R, al estar almacenados en rutas específicas de la aplicación son fácilmente accesibles desde la misma.

Por ultimo para la ejecución de los scripts de R bajo demanda desde Python, se ha optado por llamar a los scripts utilizando el módulo “os”, que provee de funciones para llamar a funciones del sistema operativo de forma portable. De esta forma Python es

capaz de hacer llamadas a los diferentes scripts con los argumentos necesarios para su funcionamiento.

6. Conclusiones

Hoy en día nos encontramos con un crecimiento imparable de las APIs, siendo pieza clave en el desarrollo de las aplicaciones webs. Como es bien conocido, una API es un conjunto de funciones que permiten a los desarrolladores utilizar determinada funcionalidad (probablemente de una aplicación externa) mediante una capa de abstracción.

A esta expansión del uso de APIs se le suma el crecimiento del uso de las redes sociales, que actualmente utiliza la inmensa mayoría de personas con acceso a internet, generando una gran cantidad de información.

Estas enormes redes sociales cuentan con sus propias APIs, con la que proveen de sus propios servicios, entre otros la compartición de parte de los datos que generan. Una de las redes sociales que destaca por la accesibilidad de su API es Twitter.

Por otra parte, la posibilidad de acceder a grandes volúmenes de información hace que lenguajes de programación estadísticos o especializados en el manejo de grandes volúmenes de datos cobren mayor relevancia en la actualidad, siendo R y Python dos ejemplos claros de estos lenguajes.

La ventaja del uso de estos lenguajes es la gran cantidad de librerías tanto de análisis como gráficas con las que cuentan y la facilidad de integración entre ellos.

En particular, algunas de las principales ventajas que han hecho que seleccione el lenguaje de programación R son:

- Proporciona una amplia gama de herramientas estadísticas y gráficas.
- Está diseñado para manejar grandes volúmenes de datos.
- Es un lenguaje interpretado, que permite su ejecución mediante comandos.
- Es un proyecto de código abierto.
- Tiene una comunidad activa.
- Es multiplataforma, funciona en los principales sistemas operativos.
- Es extensible, se puede ampliar su funcionalidad mediante el uso de librerías y paquetes.

Todo esto ha permitido desarrollar un sistema híbrido capaz de comunicarse con la API de Twitter, aplicando técnicas de Visual Analytics, para obtener la información necesaria, procesarla y analizarla apoyándose en librerías propias del lenguaje R, y a su vez comunicarse con una aplicación web desarrollada en Python que permite el acceso y explotación de los resultados.

Dicho sistema permite realizar el análisis, sobre un término seleccionado, de las tendencias, la evolución de la participación, las relaciones entre los usuarios participantes, los sentimientos expresados por estos y el rol adoptado.

Visual Analytics posee una gran variedad de herramientas para sus diferentes fases, entre otros:

- Adquisición de datos: Junar, ScraperWiki.
- Filtrado y conversión de formatos: Google Refnie, Data Wrangler.
- Análisis: R, Gephi, Google Fusion Tables.
- Publicación: HTML5, Adobe Flash.

En este proyecto, se han utilizado las herramientas de Visual Analytics: Librería Rtweet, para la obtención de los datos, PostgreSQL para el almacenamiento de los datos, el lenguaje de programación R para la fase de análisis y el framework Django con el lenguaje de programación Python, con HTML5, CSS y JavaScript para la visualización de datos.

Estas herramientas han sido eficaces para el objetivo del proyecto, siendo capaces de gestionar grandes cantidades de información. Además, de proporcionar una fácil integración.

En mi opinión, el desarrollo de este sistema ha requerido el uso de múltiples tecnologías, permitiéndome aprender un poco sobre cada una de ellas, dándome una visión general sobre los diferentes sistemas involucrados en una plataforma web.

Este trabajo me ha ayudado a consolidar lo aprendido durante el transcurso del grado y al tiempo ha ampliado el nivel de conocimiento en otras materias.

7. Bibliografía

7.1	Referencias bibliográficas	64
7.2	Referencias web.....	64

7.1 Referencias bibliográficas

IAB Spain (2018) Estudio anual redes Sociales 2018

Congosto Martínez, M. L. (2016) Caracterización de usuarios y propagación de mensajes en Twitter en el entorno de temas sociales.

Santana, J.S., Farfán E.M. (2014) EL arte de programar en R: un lenguaje para la estadística. Instituto Mexicano de Tecnología del Agua. UNESCO. Comité Nacional mexicano del Programa Hidrológico Internacional.

Kohlhammer, J., et al. (2011) Procedia Computer Science 7, 117–120

Kashnitsky, Y., (2014) Visual Analytics in FCA-based Clustering, Higher School of Economics. Moscow. Russia

McGhee, G., (2011), Tooling Up for Digital Humanities Seminar.

7.2 Referencias web

ABC, (2018), Facebook, Twitter, Instagram y WhastApp: a la caza del usuario, https://www.abc.es/tecnologia/moviles/aplicaciones/abci-facebook-twitter-instagram-y-whastapp-cuantos-seguidores-tienen-201802150936_noticia.html

IEBS, ¿Que es el Visual Analytics?-Visual Analytics en Business Intelligence & Big DataVisual Analytics en Business Intelligence & Big Data. <https://comunidad.iebschool.com/visualanalyticsbusinessintelligencebigdata/que-es-el-visual-analytics/>

Mozilla, Introducción a Django, <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducción>

Colorlib, ColorlibHQ/CoolAdmin: Free Bootstrap 4.1 based admin dashboard template, <https://github.com/ColorlibHQ/CoolAdmin>

8. Anexo

Manual de usuario

Contenido

1.	Instalación	69
1.1.	Instalación	69
2.	Descripción general del sistema.....	70
2.1.	Pantalla inicial	70
2.2.	Barra de navegación	70
2.3.	Tendencias.....	71
2.4.	Evolución.....	72
2.5.	Relaciones.....	72
2.6.	Sentimientos.....	73
2.7.	Usuarios	74
2.8.	Tendencias del momento	75
3.	Sección del gestor.....	76
3.1.	Acceso.....	76
3.2.	Gestión del histórico de análisis	76
3.3.	Creación de un nuevo análisis	77
4.	Sección del administrador	78
4.1.	Acceso.....	78
4.2.	Barra de navegación de administración	79
4.3.	Página inicial de administración	79
4.4.	Edición de tablas	80
4.5.	Edición de datos	81

1. Instalación

1.1. Instalación

Para el despliegue de la aplicación es necesario instalar previamente lo siguiente:

- Instalar las librerías para la compilación y ejecución de R, en concreto la versión 3.5.0.
- Instalar Python 3.6.6.
- El sistema gestor de bases de datos PostgreSQL en su versión 10.5.
 - Crear un usuario con los permisos necesarios.
 - Crear una base de datos para alojar los datos de la aplicación.
- Instalar el framework Django, en la versión 2.1, esto requiere de los siguientes pasos:
 - Crear un entorno virtual (virtual env, que contendrá los paquetes de Python usados en el proyecto).
- Instalar el entorno de desarrollo integrado, IDE (opcional).

Para el uso de los comandos de Django desde la consola se ha de activar el entorno virtual con el comando `workon 'nombreDelEntornoCreado'`.

Para usar Django con una base de datos PostgreSQL se debe instalar el paquete de Python `psycopg2`, he incluir los datos de la conexión con la base de datos en el fichero `setting.py` dentro de la estructura del proyecto.

Para el correcto funcionamiento de los scripts de R se debe añadir los datos de conexión con la base de datos en estos.

Se debe añadir los datos correspondientes a la autenticación en la API de Twitter en el script de R para obtención de datos “`CargaTweets1.R`”.

2. Descripción general del sistema

2.1. Pantalla inicial

Al acceder al sistema, se visualizará la página inicial, en la cual encontraremos la lista de los análisis disponibles (1).

Para acceder a los resultados de un análisis se selecciona el cuadro de la lista correspondiente a dicho análisis (2).



Figura 19: Pantalla de inicio

2.2. Barra de navegación

En la parte superior de la aplicación se encuentra la barra de navegación principal, dividida en 3 secciones:

- (1) Experimentos: seleccionando la opción de experimentos se accederá a la página inicial, en la que se podrá volver a seleccionar un análisis para visualizar.
- (2) Enlaces: en la segunda sección se podrá acceder a los distintos resultados de un análisis. Si no se ha seleccionado alguno previamente se visualizarán los resultados del último análisis realizado.

- (3) Acceso administrador: seleccionando la opción “¿eres gestor?”, se accederá al formulario de acceso correspondiente.

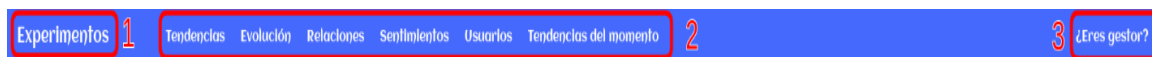


Figura 20: Barra de navegación

2.3. Tendencias

Seleccionando la opción “Tendencias” en la barra de navegación (1), se accede a la visualización de los análisis de la tendencia. En esta sección se distinguen las siguientes partes:

- (2) Termino del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.
- (3) Titulo de la sección y numero de palabras obtenidas en los tweets.
- (4) Grafico de barras que muestra el número de apariciones de las palabras más relevantes.
- (5) Nube de palabras que representa las palabras más usadas por los usuarios.

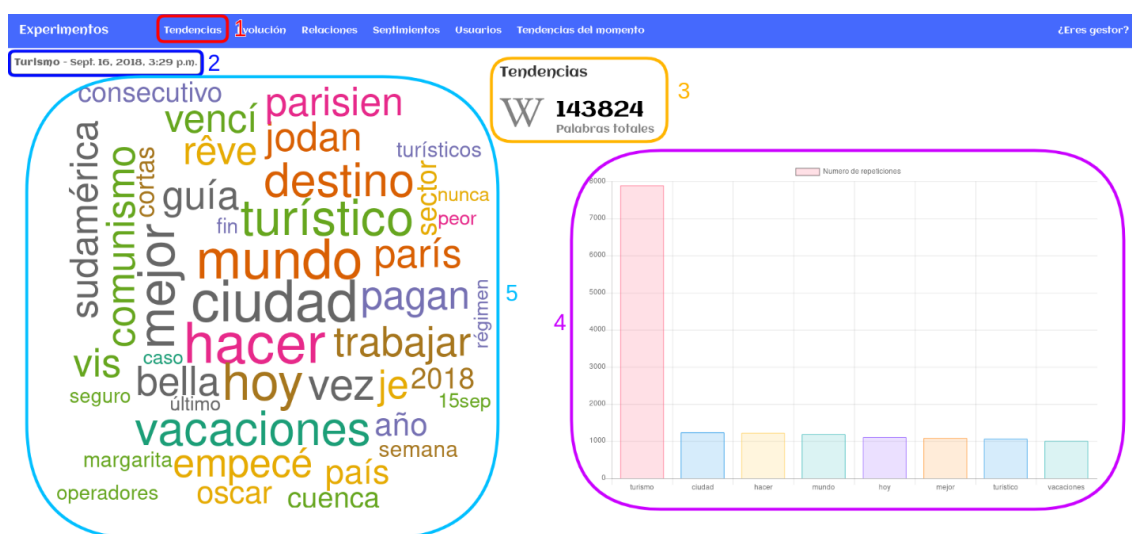


Figura 21: Página de tendencias

2.4. Evolución

Seleccionando la opción “Evolución” en la barra de navegación (1), se accede a la visualización de los análisis de la evolución. En esta sección se distinguen las siguientes partes:

(2) Termino del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.

(3) Nombre de la página y el número de tweets procesados.

(4) Gráfica que muestra la evolución del termino estudiado.

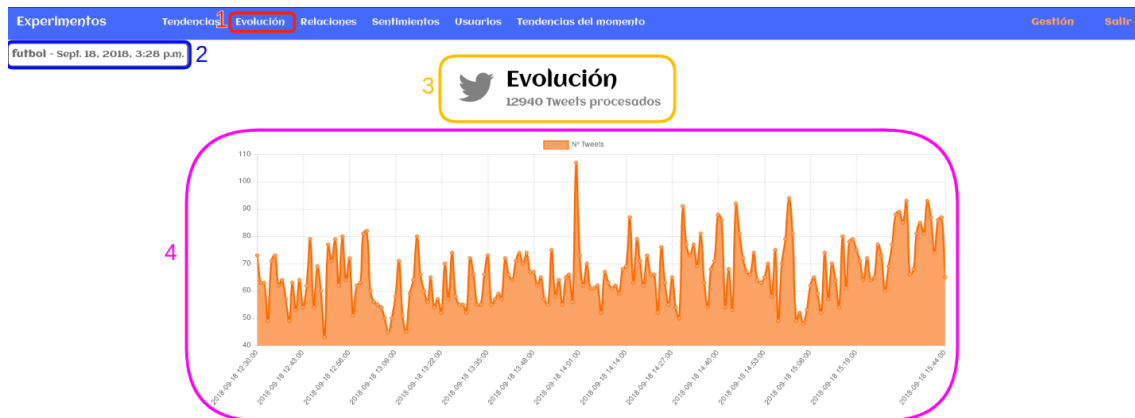


Figura 22: Página de evolución

2.5. Relaciones

Seleccionando la opción “Relaciones” en la barra de navegación (1), se accede a la visualización de los análisis de las relaciones entre usuarios. En esta sección se distinguen las siguientes partes:

(2) Termino del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.

(3) Grafo que representa las relaciones entre usuarios extraídas de los tweets. Dicho gráfico es interactivo, permitiendo al usuario seleccionar un nodo del

grafo, de forma que se resaltarán el nodo elegido y los directamente conectados con él.

- (4) Selección del tipo de relación a representar. Seleccionando una de las 3 opciones disponibles el usuario podrá alternar en el tipo de relación que tendrá el grafo mostrado.

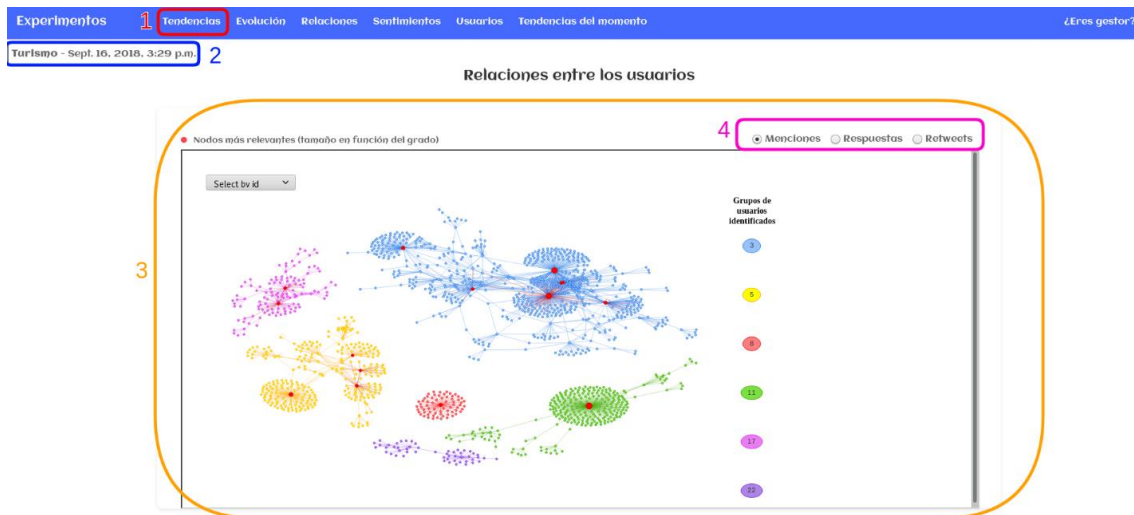


Figura 23: Página de relaciones entre usuarios

2.6. Sentimientos

Seleccionando la opción “Sentimientos” en la barra de navegación (1), se accede a la visualización de los análisis de los sentimientos expresados por los usuarios. En esta sección se distinguen las siguientes partes:

(2) Término del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.

(3) Número de usuarios totales analizados.

(4) Número total de palabras analizadas.

(5) Grafo que expresa los sentimientos de los usuarios en las distintas comunidades.

(6) Gráfica de la distribución de los sentimientos expresados por los usuarios

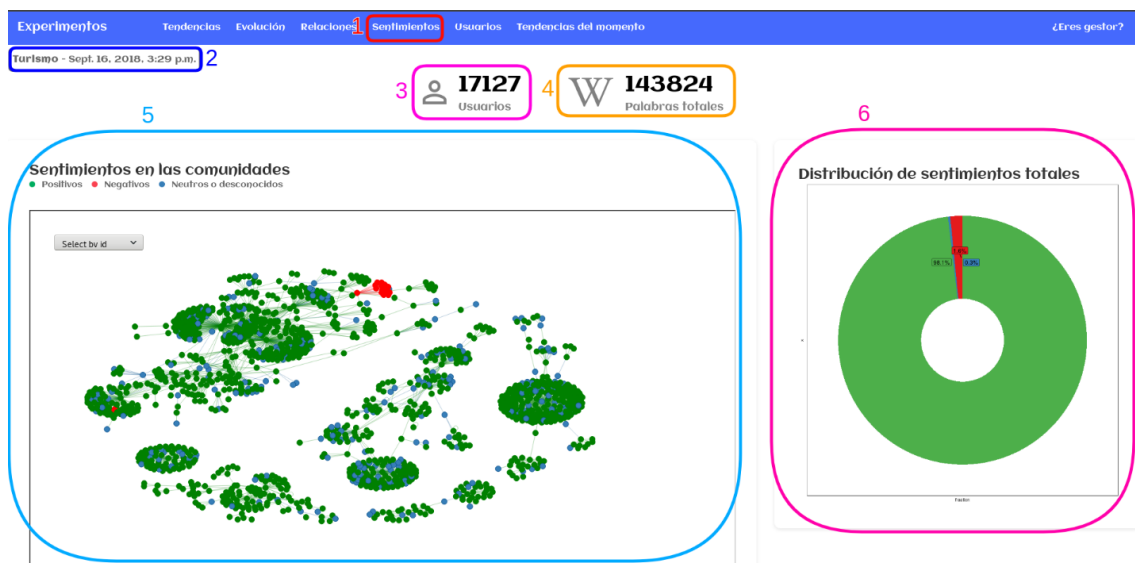


Figura 24: Página de sentimientos

2.7. Usuarios

Seleccionando la opción “Usuarios” en la barra de navegación (1), se accede a la visualización de los datos de los usuarios analizados. En esta sección se distinguen las siguientes partes:

(2) Termino del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.

(3) Tabla con los datos de los usuarios. Para ordenar por los valores de cualquier columna se debe seleccionar la cabecera de dicha columna.

(4) Cuadro de búsqueda. Para buscar cualquier dato en la tabla basta con introducir la información a buscar en el cuadro de búsqueda.

(5) Paginación: Se podrá avanzar o retroceder de página en la tabla seleccionando la opción apropiada en la sección de paginación de la tabla.

Experimentos

Tendencias

Evolución

Relaciones

Sentimientos

Usuarios

Tendencias del momento

¿Eres gestor?

Turismo - Sept. 16, 2018, 3:29 p.m.

2

Información de los usuarios

4

Busqueda:

3

Nombre	Seguidores	Tweets	Siguiendo	Me gusta	Creación de la cuenta	Clasificación
00E7D247C2E64DC	25	1.118	163	19	2014-06-16 02:16:52	Retulteador
0110ROJ9SIO11	173	17.601	188	15.522	2016-08-19 23:35:39	Común
011M9B	44	1.375	15	1.318	2013-06-25 15:50:45	Retulteador
01MILY	435	55.455	150	114	2009-12-14 14:13:30	Común
04040600T	3.307	176.983	4.966	134.608	2015-09-15 19:23:44	Común
0412_9952057	87	7.050	148	3.182	2017-10-25 17:31:33	Común
04H9RIM	213	3.464	130	79	2011-04-11 21:38:40	Ultavoz
06_SHIRO	245	9.680	587	4.920	2017-05-02 14:51:46	Común
07MIKE	343	26.248	398	11.206	2009-05-12 22:46:42	Común
0PIOJ9	12	87	227	4	2018-07-06 03:16:12	Común

Mostrando desde 1 a 10 de 6.187 entradas

5

anterior

1

2

3

4

5

...

619

Siguiente

Figura 25: Página de usuarios

2.8. Tendencias del momento

Seleccionando la opción “Tendencias del momento” en la barra de navegación (1), se accede a la visualización de los datos referentes a los temas con mayor relevancia a nivel mundial y nacional en el momento en el que se realizó el análisis. En esta sección se distinguen las siguientes partes:

(2) Termino del análisis del que se visualizan los resultados y la fecha en la que se realizó el análisis.

(3) Tabla con las tendencias a nivel mundial.

(4) Tabla con las tendencias a nivel nacional.

Experimentos

Tendencias

Evolución

Relaciones

Sentimientos

Usuarios

Tendencias del momento

¿Eres gestor?

Turismo - Sept. 16, 2018, 3:29 p.m.

Tendencias del momento

Tendencias mundiales

Douglas Costa

Nº DE TWEETS: 16,828

#JuveSassuolo

Nº DE TWEETS: 50,156

#SingaporeGP

Nº DE TWEETS: 118,366

#Gr'sYoungJaeDay

Nº DE TWEETS: 112,344

#FeltzDomlogo

Nº DE TWEETS: 29,881

#MulheresContraOBolsonaro

Tendencias en España

#GHVIP16S

Nº DE TWEETS: 59,533

Douglas Costa

Nº DE TWEETS: 16,828

#EspañayolLevante

Sassuolo

Nº DE TWEETS: 32,430

#SingaporeGP

Nº DE TWEETS: 118,366

#TJCopeMáster

Figura 26: Página de tendencias del momento

3. Sección del gestor

3.1. Acceso

Un usuario con el perfil gestor podrá acceder a las opciones propias de su rol rellenando el formulario de autenticación correspondiente (1) y pulsando la opción “Acceder”.



Figura 27: Acceso gestor.

3.2. Gestión del histórico de análisis

En la página principal de la sección del gestor se mostrará el siguiente contenido:

- (1) Gestión: Enlace a la página principal de gestión. Con este enlace el gestor podrá volver a la pantalla actual desde cualquier otra vista.
- (2) Salir: Seleccionando esta opción se cerrará la sesión actual.
- (3) Tabla de histórico de análisis: Tabla con los datos de los análisis realizados.

En esta el gestor podrá:

- a. Editar la descripción de un análisis editando el respectivo campo en la tabla (6).
- b. Activar o desactivar un análisis realizado (7).
- c. Borrar definitivamente un análisis (8).

Experimentos

Tendencias

Evolución

Relaciones

Sentimientos

Usuarios

Tendencias del momento

Gestión

Salir

3

4

Histórico

Busqueda actual

5

1

2

Término	Fecha	Descripción	Estado	Activo	Borrar
fútbol	18-09-2018 15:28:44	Busqueda sobre fútbol	Completada	✓	
psoe	18-09-2018 15:28:32	Busqueda sobre psoe	Completada	✓	
pp	17-09-2018 12:02:11	Busqueda sobre pp	Completada	✓	
Barcelona	16-09-2018 15:59:42	Busqueda sobre Barcelona	Completada	✓	
Málaga	16-09-2018 15:47:25	Busqueda sobre Málaga ejemplo	Completada	✓	
Turismo	16-09-2018 15:29:30	Busqueda sobre turismo en España	Completada	✓	

Anterior

1

Siguiente

Figura 28: Histórico de análisis

Para eliminar definitivamente un análisis el gestor tras seleccionar la opción de eliminar deberá confirmar la operación pulsando la opción correspondiente.

Histórico

Busqueda actual

Término	Fecha	Descripción	Estado	Activo	Borrar
fútbol	18-09-2018 15:28:44	Búsqueda sobre fútbol	Completada	✓	
psoe	18-09-2018 15:28:32	Búsqueda sobre PSOE	Completada	✓	
pp	17-09-2018 12:02:11		Completada	✓	
Barcelona	16-09-2018 15:59:42		Completada	✓	
Málaga	16-09-2018 15:47:25		Completada	✓	
Turismo	16-09-2018 15:29:30		Completada	✓	

¿Estas seguro?

Una vez borrado no se podrá recuperar

Cancel

OK

Anterior

1

Siguiente

Figura 29: Confirmación de borrado

3.3. Creación de un nuevo análisis

En la página principal de la sección del gestor, seleccionando la pestaña de “Búsqueda actual” se mostrará el siguiente contenido:

- (1) Sección actual.
- (2) Término a analizar.

- (3) Numero de tweets objetivo
- (4) Descripción opcional del análisis.
- (5) Botón para realizar el análisis.

El usuario gestor, rellendo el formulario correspondiente (1-4) y pulsando el botón para comenzar el análisis (5) lanzará el proceso para la obtención de datos y el análisis correspondiente.

The screenshot shows a web interface with a blue header bar containing navigation links: 'Experimentos', 'Tendencias', 'Evolución', 'Relaciones', 'Sentimientos', 'Usuarios', and 'Tendencias del momento'. On the right side of the header are 'Gestión' and 'Salir' links. Below the header, there are two tabs: 'Historico' and 'Búsqueda actual', with the latter being selected and highlighted with a red box and the number 1. The main content area is titled 'Búsqueda actual' and contains a form with the following elements: a 'Término' input field with 'futbol' (highlighted with a red box and number 2), a 'Número de tweets' input field with '7000' (highlighted with a blue box and number 3), a 'Descripción' text area with 'Búsqueda sobre futbol' (highlighted with an orange box and number 4), and an 'Analizar' button (highlighted with a pink box and number 5).

Figura 30: Nueva búsqueda

4. Sección del administrador

4.1. Acceso

Un usuario con el perfil administrador podrá acceder a las opciones propias de su rol rellendo el formulario de autenticación correspondiente (1) y pulsando la opción “Log in”.

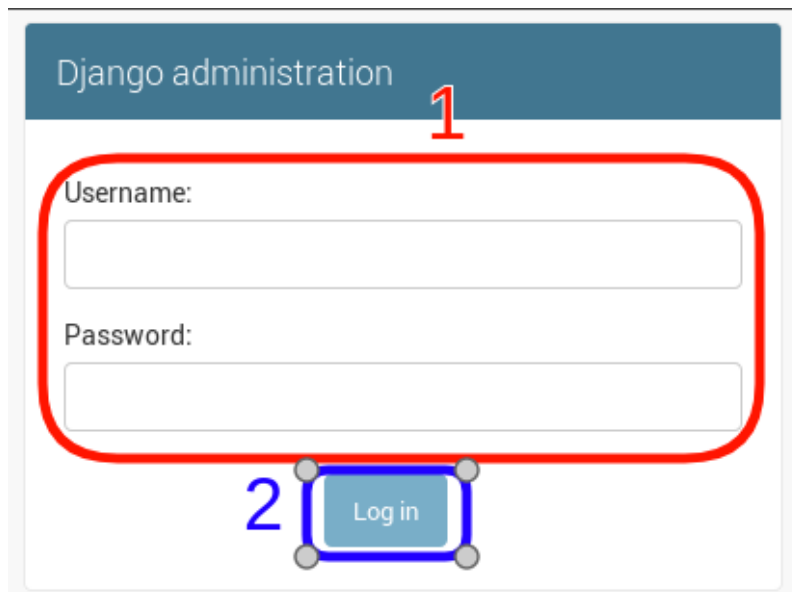


Figura 31: Acceso administrador

4.2. Barra de navegación de administración

En las páginas reservadas al administrados se mostrará un menú de navegación con las siguientes opciones:

- (1) Acceso directo a la página principal de la aplicación.
- (2) Opción para cambiar su propia contraseña.
- (3) Opción para cerrar la sesión.



Figura 32: Menú de navegación de Administración

4.3. Página inicial de administración

En la página principal de administración aparecerá las siguientes opciones:

- (1) Tablas: Seleccionando la tabla oportuna se accederá a la vista para la edición de esta.
- (2) Seguridad: En esta sección se podrá editar los usuarios o grupos.

(3) Acciones recientes: muestra un historial con las últimas acciones realizadas.



Figura 33: Opciones de administración

4.4. Edición de tablas

En la página de Edición de tablas un administrador podrá seleccionar una de las entradas para editarla (1) o crear una nueva (2):



Figura 34: Lista de entradas

4.5. Edición de datos

Al seleccionar una entrada editar los datos correspondientes (1), tras lo que deberá almacenar los cambios seleccionando la opción correspondiente (2), o borrar la entrada (3).

Home - Analytics - Historico de búsquedas - futbol - 2018-09-18 - 15:28:43

Change search_history

Search: futbol

☒ Active

Description: Búsqueda sobre futbol

NTweets: 12940

☒ Complete

Delete

Save and add another Save and continue editing Save

Figura 35: Gestión de una entrada